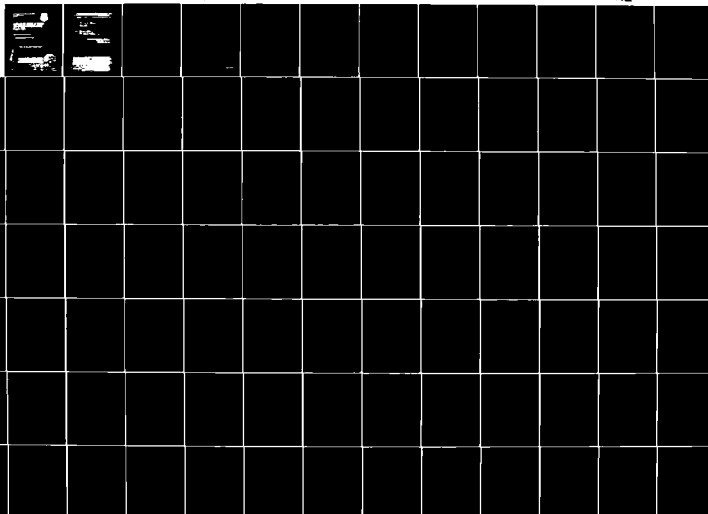


AD-A118 498

MCDONNELL DOUGLAS RESEARCH LABS ST LOUIS MO F/G 9/2
ALGORITHM FOR SURFACE OF TRANSLATION ATTACHED RADIATORS (A-STAR--ETC(U))
MAY 82 L N MEDGYESI-MITSCHANG, J M PUTNAM F30602-80-C-0106
RADC-TR-82-113-VOL-3 NL

UNCLASSIFIED

1 of 3
A-118 498



AD A118498



TABLE OF CONTENTS

	<u>Page No.</u>
LISTING OF CODES.....	1
BOTSEG.....	1
BOTZSS.....	7
BOTZSC.....	21
BOTZCC.....	35
BOTZSW.....	53
BOTZWW.....	69
BOTZCW.....	85
BOTINV.....	101
BOTINVA.....	113
BOTAC.....	129
BOTRA.....	147
BOTSCM.....	187
BOTSCB.....	211

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	☐
By	
Distribution	
Availability	
Dist	
A	



LISTING OF CODES

1

AAAAAAAAAA 00000000 TTTTTTTTTT SSSSSSSS EEEEEEEEE EEEEEEEEE
AAAAAAAAAA 00000000 TTTTTTTTTT SSSSSSSS EEEEEEEEE EEEEEEEEE
AA 0000 0000 TTTT SSSSSSSS EEEEEEEEE EEEEEEEEE
AA 00 00 00 TT SSSSSSSS EEEEEEEEE EEEEEEEEE
AA 00 00 00 TT SSSSSSSS EEEEEEEEE EEEEEEEEE
AA 00 00 00 TT SSSSSSSS EEEEEEEEE EEEEEEEEE
AA 00 00 00 TT SSSSSSSS EEEEEEEEE EEEEEEEEE
AA 00 00 00 TT SSSSSSSS EEEEEEEEE EEEEEEEEE
AA 00 00 00 TT SSSSSSSS EEEEEEEEE EEEEEEEEE
AA 00 00 00 TT SSSSSSSS EEEEEEEEE EEEEEEEEE
AA 00 00 00 TT SSSSSSSS EEEEEEEEE EEEEEEEEE
AA 00 00 00 TT SSSSSSSS EEEEEEEEE EEEEEEEEE
AA 00 00 00 TT SSSSSSSS EEEEEEEEE EEEEEEEEE
AAAAAAAAAA 00000000 TTTTTTTTTT SSSSSSSS EEEEEEEEE EEEEEEEEE
AAAAAAAAAA 00000000 TTTTTTTTTT SSSSSSSS EEEEEEEEE EEEEEEEEE

```

C014C      REAL XB(83),YR(83)
C0110      REAL XTAB(100),YTAB(100),XC(100),YC(100)
C0120      READ(5,2) NPTS,NP
C0130      2      FORMAT(2I3)
C0140C      READ BODY COORDINATES.
C0150C
C0160C
C0170C      DO 100 I=1,NPTS
C0180C          XC(I)=0.0
C0190C          YC(I)=0.0
C0200C      100  READ(5,3) XTAB(I),YTAB(I),XC(I),YC(I)
C0210C      3      FORMAT(4E10.4)
C0220C
C0230C      SEGMENT THE BODY INTO NP-1 SEGMENTS OF EQUAL LENGTH.
C0240C
C0250C          CALL INTRPL(NP-1,NPTS,P,XTAB,YTAB,XC,YC,XB,YB)
C0260C          WRITE(6,1) P,NP
C0270C      1      FORMAT(17,12H PERIMETER =,F10.4,/,
C0280C          1 5H NP =,I4)
C0290C          WRITE(6,33) (YR(I),I=1,NP)
C0300C          WRITE(6,34) (XR(I),I=1,NP)
C0310C      33      FORMAT(17,3H YR,/, (1X,10F10.4))
C0320C      34      FORMAT(17,3H XR,/, (1X,10F10.4))
C0330C
C0340C      PLOT THE CALCULATED BODY COORDINATES.
C0350C
C0360C          CALL PLOTB(XB,YR,NP,31)
C0370C          STOP
C0380C      END
C0390C *****
C0400C      SUBROUTINE INTRPL(NP,NPTS,P,XTAB,YTAB,XC,YC,XR,YR)
C0410C
C0420C          THIS SUBROUTINE WILL TAKE BODY COORDINATES (XTAB(I),YTAB(I),I=1,NP)
C0430C          AND CALCULATE EQUALLY SPACED BODY COORDINATES (XR(I),YR(I),I=1,NP+1)
C0440C          THE POINTS XTAB AND YTAB ARE ASSUMED TO BE EITHER
C0450C          1) ENDPPOINTS OF STRAIGHT LINES OR 2) ENDPPOINTS OF CIRCULAR ARCS.
C0460C          FOR EXAMPLE IF EITHER XC(K) OR YC(K) IS NOT ZERO, THEN
C0470C          (XTAB(K),YTAB(K)) AND (XTAB(K+1),YTAB(K+1)) ARE ENDPPOINTS OF THE
C0480C          CIRCULAR ARC WITH CENTER (XC(K),YC(K)). NOTE, IF THE RADIUS
C0490C          IS DIFFERENT AT THE ENDPPOINTS, IT IS ASSUMED TO HAVE A LINEAR
C0500C          DEPENDENCE WITH ANGLE BETWEEN THE ENDPPOINTS. IF XC(K)-YC(K)=0,
C0510C          IT IS ASSUMED TO BE STRAIGHT.
C0520C          P IS RETURNED AS THE PERIMETER OF THE BODY.
C0530C
C0540C          REAL XTAB(1),YTAB(1),XR(1),YR(1)
C0550C          REAL XC(1),YC(1)
C0560C          REAL MOVE
C0570C          P=0.0
C0580C          K=NP-1
C0590C          DO 25 I=1,K
C0600C              IF (XC(I).NE.0.0 .OR. YC(I).NE.0.0) G7 TO 20
C0610C              X=XTAB(I+1)-XTAB(I)

```

```

C2060      Y=YTAB(I+1)-YTAB(I)
C2070      P=P+SORT(X*X+Y*Y)
C2080      GO TO 25
20      X=XTAB(I)-XC(I)
      Y=YTAB(I)-YC(I)
      R=SQRT(X*X+Y*Y)
      X1=XTAB(I+1)-XC(I)
      Y1=YTAB(I+1)-YC(I)
      R1=SQRT(X1*X1+Y1*Y1)
      THETA=ACOS((X*X1+Y*Y1)/(R*R1))
      P=P+(R*R1)*THETA/2.0
25      CONTINUE
      PI=P/PI
      XR(I)=XTAB(I)
      YR(I)=YTAB(I)
      XR(NI+1)=XTAB(NP)
      YR(NI+1)=YTAB(NP)
      NK=NK+1
      MOVE=C.C
      DO 200 I=1,K
      X1=XTAB(I)
      Y1=YTAB(I)
      IF(XC(I).NE.0.0 .OR. YC(I).NE.0.0) GO TO 40
      X=XTAB(I+1)-X1
      Y=YTAB(I+1)-Y1
      O=SQRT(X*X+Y*Y)
      GO TO 30
40      X=X1-XC(I)
      Y=Y1-YC(I)
      R=SQRT(X*X+Y*Y)
      X1=XTAB(I+1)-XC(I)
      Y1=YTAB(I+1)-YC(I)
      R1=SQRT(X1*X1+Y1*Y1)
      THETA=ACOS((X*X1+Y*Y1)/(R*R1))
      O=(R+R1)*THETA/2.0
      THETA=ATAN2(Y,X)
      THETA1=ATAN2(Y1,X1)
      DIR=+1 FOR CLOCKWISE DIRECTION
      DIR=-1 FOR COUNTER-CLOCKWISE DIRECTION.
      DIR=+1
      IF(THETA.GT.THETA1 .AND. THETA-THETA1.GT.3.14159266) DIR=-1.
      IF(THETA.LT.THETA1 .AND. THETA1-THETA.LT.3.14159265) DIR=-1.
      IF(MOVE.D.LT.PI) GO TO 200
50      NK=NK+1
      IF(XC(I).NE.0.0 .OR. YC(I).NE.0.0) GO TO 60
      XR(NK)=X1+(PI-MOVE)*(XTAB(I+1)-X1)/O
      YR(NK)=Y1+(PI-MOVE)*(YTAB(I+1)-Y1)/O
      GO TO 70
60      X=X1-XC(I)
      Y=Y1-YC(I)
      R=SQRT(X*X+Y*Y)
      THETA=ATAN2(Y,X)

```

```

C2510      R=R*(PI-MOVE)*(R1-R)/D
C2520      XR(NK)=XC(I)+R*COS(THETA-DIR*(PI-MOVE)/R)
C2530      YR(NK)=YC(I)+R*SIN(THETA-DIR*(PI-MOVE)/R)
C2540 7C    IF(NK.EQ.NI) RETURN
C2550      D=D-PI+MOVE
C2560      MOVE=C.O
C2570      XI=XR(NK)
C2580      YI=YR(NK)
C2590      GO TO 50
C2600 2C    MOVE=MOVE+D
C2610      RETURN
C2620      END
C2630      *****
C2640 1C    SUBROUTINE PLOTB(X,Y,N,NR)
C2650      *****
C2660      WRITTEN 2/14/74    BY J. M. PUTNAM    DEPT 220    X23877
C2670      THIS ROUTINE PRODUCES A LINEAR XY PLOT.
C2680      N IS THE NUMBER OF POINTS TO BE PLOTTED.
C2690      NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
C2700      NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
C2710      NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
C2720
C2730      REAL X(1),Y(1),HEAD(10)
C2740      INTEGER LINE(10),BLANK,STAR
C2750      DATA BLANK,STAR /1H,1H+/
C2760      NC=51
C2770      N10=(NC-1)/10
C2780      WRITE(108,500)
C2790 5C    FORMAT(//,17H BODY COORDINATES)
C2800      WRITE(108,504)
C2810      XMIN=X(1)
C2820      XMAX=X(1)
C2830      YMIN=Y(1)
C2840      YMAX=Y(1)
C2850      DO 6 I=1,N
C2860      IF(X(I).LT.XMIN) XMIN=X(I)
C2870      IF(X(I).GT.XMAX) XMAX=X(I)
C2880      IF(Y(I).LT.YMIN) YMIN=Y(I)
C2890      IF(Y(I).GT.YMAX) YMAX=Y(I)
C2900 6      CONTINUE
C2910      DEL=XMAX-XMIN
C2920      IF(YMAX-YMIN.GT.DEL) DEL=YMAX-YMIN
C2930      XMAX=XMIN+DEL
C2940      YMAX=YMIN+DEL
C2950      DO 5 I=1,N10
C2960      Z=I
C2970 5      HEAD(I)=(XMAX-XMIN)*Z/N10+XMIN
C2980      DY=(YMAX-YMIN)/(NR-1)
C2990      Z=YMAX+DY
C3000      YL=Z-DY/2.

```

```

C3325      IBLK=0
C3330      DO 7 J=1,NR
C3340      DO 8 K=1,NC
C3350 8      LINE(K)=BLANK
C3360      Z=Z-DY
C3370      YU=YI
C3380      YL=Z-DY/2.
C3390      DO 9 I=1,N
C3400      IF(Y(I).GE.YU) GO TO 9
C3410      IF(Y(I).LT.YL) GO TO 9
C3420      IBLK=1
C3430      K=(X(I)-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
C3440      IF(K.GT.NC) K=NC
C3450      LINE(K)=STAR
C3460 9      CONTINUE
C3470      IF(IBLK.NE.0) WRITE(108,508) Z,(LINE(K),K=1,NC)
C3480 7      CONTINUE
C3490      WRITE(108,504)
C3500      WRITE(108,3002)
C3510      WRITE(108,507) XMIN,(HEAD(I),I=1,N10)
C3520      RETURN
C3530 504  FORMAT (1X,14(1H-),1H.,10(5H----.),1H-)
C3540 507  FORMAT(10X,11(F10.4))
C3550 508  FORMAT(1X,F12.4,1X,1HI,51A1,1HI)
C3560 3002 FORMAT(4X,7H R / 2 ,4X,1HI,5(9X,1HI))
C3570      END
C3580
C3590

```

7

PRECEDING PAGE B

```

0010080TZ,P30,T100,CH14C000,STCK3.
00110ACCOUNT,NO3265,BOT2A
00120BANNERS(OUTPUT)*J. PUTNAM*DEPT 220*BLO 110-4**
00130DEFINE(OUTFIL=ZSSX)
00138FINIR=0,OPT=2)
00140IDSET(PRESET=INDEF)
00150LEGO.
00210EXIT.
00230/EDR
00240      PROGRAM BOTZSS(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
00250      1 OUTFIL,TAPE7=OUTFIL)
00252C
00254C BOTZSS GENERATES THE BOT Z MATRIX.
00256C
00258C UNIT 5 IS THE CARD READER.
00260C UNIT 6 IS THE LINE PRINTER.
00262C UNIT 7 IS A DISK FILE FOR OUTPUT OF THE Z MATRIX.
00264C
00280      COMPLEX A3,Z(6400),G(3403),GB(3403),U
00281      COMPLEX GS,GRS
00282      DIMENSION YGHT(40),ZMGHT(40)
00283      COMMON /WAVE/ RK
00284      COMMON /BOT1/ NMODE,NPT,NBAND
00285      COMMON /BOT2/ NP,BL,YB(83),YB(83),YB1(82),XB1(82)
00286      COMMON /BOT3/ DH(82),SV(82),CV(82)
00287      COMMON /BOT5/ T(160),TP(160),TZ(160)
00290      COMMON /BOT6/ TEDGF,IUNIF
00320      COMMON /INT/ M,N,RKL,RHO?,DELTA
00325      COMMON /UCONST/ AMN,ABMN,PTN,PIN
00326      DIMENSION LI(2),LJ(2),NAME(2),NAMJ(2)
00327      EXTERNAL FUNC,FUNCB,FUNCS,FUNCS
00328      DATA NAME,NAMJ /4HS(T),4HS(Z),4HS(T),4HS(Z)/
00430      U=(0.,1.)
00432      PT=3.14159265
00434      ETA=376.707
00435      READ(5,1) BK
00436 1      FORMAT(E15.7)
00437      WRITE(6,2) BK
00438 2      FORMAT(9H1      RK,/,E15.7)
00510C
00512C CALL BOTIN
00520C
00530      WRITE(7) NP
00790      KG=NP-1
00810      NM=(NP-3)/2
00820      NM2=NM*2
00832      LSS=NP2*NM2
00833      NG=KG*KG
00834      BKL=BK*BL
00835      BK2=BK*BK
00836C CONSTANTS USED IN SUMMATION OF IMPEDANCE ELEMENTS.
00840      C1=BK*ETA

```

```

00841      C2=PI*ETA/BK/RL
00842      C3=C2
00843      C4=C1
00845      B4=(PI/BK/BL)**2
00890C    *** END INITIALIZATION OF PARAMETERS ***
00892C    CALCULATE IMPEDANCE LOADING WEIGHTS ON 9DT.
00894      DO 10 J=1,NM
00896          J2=2*(J-1)+1
00898          J3=J2+1
00900          J4=J3+1
00902          J5=J4+1
00904          J6=4*(J-1)+1
00906          J7=J6+1
00908          J8=J7+1
00910          J9=J8+1
00912          TWGHT(J)=2.*BL*(DH(J2)*T(J6)+DH(J3)*T(J7)+
00914          1 DH(J4)*T(J8)+DH(J5)*T(J9))
00916          ZWGHT(J)=2.*BL*(DH(J2)*TZ(J6)+DH(J3)*TZ(J7)+
00918          1 DH(J4)*TZ(J8)+DH(J5)*TZ(J9))
00920      10 CONTINUE
00922          WRITE(7) (TWGHT(J),J=1,NM)
00924          WRITE(7) (ZWGHT(J),J=1,NM)
00939C    COMPUTATION OF IMPEDANCE MATRICES Z(N,N), WHERE M = 0 TO NM0DE-1
00940C    AND N = -M TO +M.
00941C    NOTE, THESE MATRICES ARE COMPUTED IN THE FOLLOWING ORDER:
00942C    1) ALONG THE MAIN DIAGONAL (M=N)
00943C    2) ALONG THE FIRST OFF-DIAGONAL (M-N=1)
00944C    3) ALONG THE SECOND OFF-DIAGONAL (M-N=2)
00945C    ETC.
00946      IM0DE=NM0DE
00947      KM0DE=2*NM0DE-1
00948C    THIS LOOP FIXES ONE OF THE DIAGONALS.
00949      DO 999 NN=1,KM0DE
00950          IF(NN.GT.NBAND) GO TO 999
00951          IF(MOD(NN,2).EQ.0) GO TO 20
00952          M=(NN+1)/2-2
00953          N=-(NN+1)/2
00954          GO TO 21
00955      20      IM0DE=IM0DE-1
00956          M=NN/2-1
00957          N=-NN/2
00958      21      CONTINUE
00959C    THIS LOOP STEPS M AND N ALONG THE FIXED DIAGONAL.
00962      DO 998 MM=1,IM0DE
00963          M=M+1
00964          N=N+1
00965          WRITE(6,54) M,N
00970      54      FORMAT(4H1 M=,I3,4H N=,I3)
00980C    CONSTANTS USED IN JMN FUNCTIONS.
00990          IF(M.EQ.N) AMN=0.5/PI
01000          IF(M.NE.N) AMN=(-1)**(M-N+1)/FLOAT(N-M)/PI/PI/2.
01010          AMN=(-1)**(M-N)/PI/2.

```

```

C1020      PIN=PI+H
C1030      PIN=PI+H
C1185C    COMPUTATION OF GREEN'S FUNCTION KERNEL FOR J.GE.I.
C1186C    NOTE: G IS A SYMMETRIC MATRIX, HENCE ONLY THE UPPER TRIANGULAR
C1187C    PORTION IS STORED BY COLUMNS FROM G(1) TO G(KG*(KG+1)/2).
C1188C    G(I,J) IS STORED IN G(I+(J-1)*J/2) WHEN I.LE.J.
C1190      I1=0
C1192      NPT=C
C1194      NINT=0
C1196      NPTS=C
C1198      NINTS=0
C1228      DO 16 J=1,KG
C1230      DO 17 I=1,J
C1240      I1=I+1
C1260      YY=Y01(J)-Y01(I)
C1270      XX=X01(J)-X01(I)
C1280      X1=ABS(XX)+ABS(YY)
C1285      IF(X1.NE.0.0) GO TO 7
C1290C    SELF TERM INTEGRATION.
C1290      IF(J.NE.1.AND. IUNIF.EQ.1) GO TO 6
C1291      DELTA=DH(I)/2./RL
C1292      CALL CSIMP(FUNCS,C.0,2.0,0.05,10,A3,G(I1),K,IER)
C1293      NPTS=NPTS+K+1
C1294      NINTS=NINTS+1
C1295      G(I1)=G(I1)*2.*BL*BL/DH(I)
C1296      GS=G(I1)
C1298      IF(N.EQ.0) GO TO 17
C1299      CALL CSIMP(FUNCS,0.0,2.0,0.05,10,A3,GB(I1),K,IER)
C1300      NPTS=NPTS+K+1
C1301      NINTS=NINTS+1
C1302      GB(I1)=GB(I1)*2.*BL*BL/DH(I)
C1303      GBS=GB(I1)
C1304      GO TO 17
C1305C    BOT SEGMENTATION IS UNIFORM(IUNIF=1).
C1306      6 G(I1)=GS
C1307      IF(N.NE.0) GB(I1)=GBS
C1308      GO TO 17
C1309C    NON-SELF TERM INTEGRATION.
C1310      7 RHDZ=(XX*XX+YY*YY)/BL/BL
C1320      CALL CSIMP(FUNC,0.0,2.0,0.05,10,A3,G(I1),K,IER)
C1322      NPT=NPT+K+1
C1324      NINT=NINT+1
C1330      G(I1)=G(I1)*BL
C1340      IF(N.EQ.0) GO TO 17
C1350      CALL CSIMP(FUNC,0.0,2.0,0.05,10,A3,GB(I1),K,IER)
C1352      NPT=NPT+K+1
C1354      NINT=NINT+1
C1360      GB(I1)=GB(I1)*BL
C1370      17 CONTINUE
C1380      16 CONTINUE
C1662      AVG=NPT/FLOAT(NINT)
C1664      AVGS=NPTS/FLOAT(NINTS)

```

```

C1666      WRITE(6,56) NINTS,AVGS,NINT,AVG
C1668 56    FORMAT(1, ' NUMBER OF INTEGRATIONS AND AVERAGE NUMBER OF ',
C1670      1, ' PCINTS PER INTEGRATION (SELF AND NON-SELF TERMS, ',
C1672      2, ' RESPECTIVELY)',/,2(I8,G10,3))
C1674C *** END CALCULATION OF GREENS'S KERNEL ***
C2005C COMPUTATION OF IMPEDANCE ELEMENTS, EQS. 17-20.
C2006C NOTE, THE SYMMETRIES IN THE SUBMATRICES OF Z(P,N) ARE USED.
C2007C THESE SYMMETRIES ARE: ZTT = TRANSPOSE(ZTT),
C2008C ZTZ = TRANSPOSE(ZZZ), AND M * TRANSPOSE(ZTZ) = -N * ZTZ.
C2010      DO 30 J=1,NM
C2020      JL=(J-1)*NM2
C2030      J3=(J-1)*4
C2040      J1=2*(J-1)
C2050      DO 31 I=1,J
C2055C COMPUTE SUBMATRIX INDICES.
C2060      L1=JL+I
C2070      L2=L1+NM
C2080      L3=NM+NM2+L1
C2090      L4=L3+NM
C2100      Z(L1)=0.
C2110      Z(L2)=0.
C2120      Z(L3)=0.
C2130      Z(L4)=0.
C2135      IF(N.EQ.0 .AND. M.EQ.0 .AND. I.EQ.J) Z(L4)=1.0
C2140      IF(ABS(I-J).GE.NPT) GO TO 28
C2150      I1=2*(I-1)
C2160      I3=I1-1
C2170      DO 70 JJ=1,4
C2180      J2=J1+JJ
C2190      J7=J3+JJ
C2200      DO 71 II=1,4
C2210      I2=I1+II
C2220      I7=I3+II
C2224C THE COMPUTATION OF J4 REFLECTS THE FACT THAT G IS SYMMETRIC,
C2225C WITH ONLY THE UPPER PORTION STORED AS NOTED ABOVE.
C2226C NOTE, IF G WAS FILLED COMPLETELY, THEN J4=(J2-1)*K6+I2.
C2230      J4=MAXC(I2,J2)
C2240      J4=(J4-1)*J4/2+MINC(I2,J2)
C2250      SS=SV(I2)*SV(J2)
C2260      CC=CV(I2)*CV(J2)
C2270C IMPEDANCE ELEMENTS COMPUTED, EQS. 17-20.
C2296C Z(L1) = ZTT, Z(L2) = ZTZ, Z(L3) = ZTZ, Z(L4) = ZZZ.
C2300      Z(L1)=Z(L1)+DH(I2)*DH(J2)*(T(I7)*T(J7)*(CC+SS)-TP(I7)*
C2310      1 TP(J7)/8K2)*G(J4)
C2320      Z(L2)=Z(L2)+DH(I2)*DH(J2)*TZ(I7)*TP(J7)*G(J4)
C2330      Z(L3)=Z(L3)+DH(I2)*DH(J2)*TP(I7)*TZ(J7)*G(J4)
C2340      IF(N.EQ.0) GO TO 71
C2350      Z(L4)=Z(L4)+DH(I2)*DH(J2)*TZ(I7)*TZ(J7)*
C2360      1 ((1.-M*N*84)*G(J4)+GR(J4))
C2370      71 CONTINUE
C2375      70 CONTINUE
C2379C MULTIPLY BY CONSTANTS, EXCEPT FOR N AND M FACTORS IN ZTZ, AND

```

```

02380C ZTZ, RESPECTIVELY,
02381      Z(L1)=U+C1*Z(L1)
02382      Z(L2)=C2*Z(L2)
02383      Z(L3)=C3*Z(L3)
02384      IF(N,NE,0) Z(L4)=U+C4*Z(L4)
02385 28 CONTINUE
02387C FILL THE Z(M,N) SUBMATRIX USING THE ZTT, ZTZ, ZTZ, AND
02388C ZZZ SYMMETRIES NOTED ABOVE.
02390C IF(I,EO,J) GO TO 29
02392C COMPUTE SUBMATRIX TRANSPOSE INOICES.
02395      L1T=(I-1)*NM2+J
02396      L2T=L1T+NM
02397      L3T=NM*NM2+L1T
02398      L4T=L3T+NM
02399      Z(L1T)=Z(L1)
02400      Z(L2T)=-M*Z(L3)
02401      Z(L3T)=M*Z(L2)
02402      Z(L4T)=Z(L4)
02404 29 Z(L2)=-M*Z(L2)
02405      Z(L3)=M*Z(L3)
02406 31 CONTINUE
02407 30 CONTINUE
02408C *** END CALCULATION OF Z(M,N) SUBMATRIX ***
02409C PRINT OUT IMPEDANCE MATRIX Z(M,N) BY SUBMATRIX BY COLUMNS.
02410      LI(1)=NM
02411      LI(2)=NM
02412      LJ(1)=NM
02413      LJ(2)=NM
02414      CALL ZLIST(2,2,LI,LJ,NAMI,NAMJ,Z)
02415C WRITE Z(M,N) TO DISK.
02416      WRITE(7) M,N
02417      WRITE(7) (Z(I),I=1,LSS)
02418 998 CONTINUE
02419 999 CONTINUE
02420 STOP
02421 END
-----
02422C SUBROUTINE BOTIN
02423C
02424C READ BOT COORDINATES AND COMPUTE BOT SEGMENT ARRAYS.
02425C
02426C COMMON /BOT1/ NM0DE,NPT,NBAND
02427C COMMON /BOT2/ NP,NL,YB(83),XB(83),YB(82),XB(82)
02428C COMMON /BOT3/ DH(82),SV(82),CV(82)
02429C COMMON /BOT5/ T(160),TP(160),TZ(160)
02430C COMMON /BOT6/ IEDGE,IUNIF
02431C READ(5,49) NM0DE,NPT,NBAND
02432C READ(5,49) NP
02433C 49 FORMAT(3I3)
02434C WRITE(6,48) NM0DE,NPT,NBAND,NP
02435C 48 FORMAT(32H NM0DE NPT NBAND NP,/,4I8)
02436C READ(5,53)(YB(I),I=1,NP)

```

```

02950      READ(5,53)(XB(I),I=1,NP)
02960      53 FORMAT(10F8.4)
02970      WRITE(6,55)
02980      55 FORMAT(/,3H YB)
02990      WRITE(6,46)(YB(I),I=1,NP)
03000      46 FORMAT(1X,10F8.4)
03010      WRITE(6,56)
03020      56 FORMAT(/,3H XB)
03030      WRITE(6,46)(XB(I),I=1,NP)
03040C PLOT THE BODY COORDINATES.
03050      CALL PLOTB(XB,YB,NP,4)
03060      READ(5,53) BL
03070      WRITE(6,47) BL
03080      47 FORMAT(/,21H HALF-LENGTH OF BOT =,F12.4)
03090C CHECK IF BOT CONTAINS AN ODD NUMBER OF POINTS.
03100      IF(MOD(NP,2).NE.1) GO TO 980
03110C CHECK IF GENERATING CURVE IS OPEN OR CLOSED.
03120      IEDGE=1
03130      IF((YB(1)-YB(NP)).NE.0..OR.(XB(1)-XB(NP)).NE.0.) GO TO 10
03140      IEDGE=0
03150      YB(NP+1)=YB(2)
03160      XB(NP+1)=XB(2)
03170      YB(NP+2)=YB(3)
03180      XB(NP+2)=XB(3)
03190      NP=NP+2
03200      WRITE(6,66) NP
03210      66 FORMAT(/,39H BOT GENERATING CURVE IS CLOSED. NP = ,I3)
03220      10 CONTINUE
03230C COMPUTATION OF BODY SEGMENT PARAMETERS.
03240      DO 57 I=2,NP
03250      I2=I-1
03260      RR1=YB(I)-YB(I2)
03270      RR2=XB(I)-XB(I2)
03280      DH(I2)=SQRT(RR1*RR1+RR2*RR2)
03290      XB(I2)=.5*(XB(I)+XB(I2))
03300      YB(I2)=.5*(YB(I)+YB(I2))
03310      SV(I2)=RR1/DH(I2)
03320      CV(I2)=RR2/DH(I2)
03330      57 CONTINUE
03340C CHECK IF BOT SEGMENTATION IS UNIFORM.
03350      IUNIF=0
03360      NP1=NP-1
03370      DO 60 I=2,NP1
03380      RR1=DH(I)/DH(I1)
03390      IF(RR1.LT.0.99..OR. RR1.GT.1.01) GO TO 20
03400      60 CONTINUE
03410      IUNIF=1
03420      WRITE(6,67)
03430      67 FORMAT(/,1 BOT GENERATING CURVE HAS UNIFORM SEGMENTATION*)
03440      20 CONTINUE
03450C COMPUTATION OF TRIANGLE FUNCTIONS T.
03460      NM=(NP-3)/2

```

```

C35C0      DO 74 J=1,NM
C35D0      J2=2*(J-1)+1
C35E0      J3=J2+1
C35F0      J4=J3+1
C35G0      J5=J4+1
C35H0      J6=4*(J-1)+1
C35I0      J7=J6+1
C35J0      J8=J7+1
C35K0      J9=J8+1
C35L0      DEL1=DM(J2)+DM(J3)
C35M0      DEL2=DM(J4)+DM(J5)
C35N0      TP(J6)=1./DEL1
C35O0      TP(J7)=1./DEL1
C35P0      TP(J8)=1./DEL2
C35Q0      TP(J9)=1./DEL2
C35R0      T(J6)=DM(J2)/2./DEL1
C35S0      T(J7)=(DM(J2)+DM(J3))/2./DEL1
C35T0      T(J8)=(DM(J4)+DM(J5))/2./DEL2
C35U0      T(J9)=DM(J5)/2./DEL2
C35V0      74 CONTINUE
C35W0      NM4=NM+4
C35X0      DO 75 J=1,NM4
C35Y0      TZ(J)=T(J)
C35Z0      75 IF(IEDGE.EQ.0) GO TO 76
C3600      TZ(1)=2.-T(1)
C3610      TZ(2)=2.-T(2)
C3620      TZ(NM4-1)=2.-T(NM4-1)
C3630      TZ(NM4)=2.-T(NM4)
C3640      76 CONTINUE
C3650      RETURN
C3660      980 WRITE(6,981)
C3670      981 FORMAT(//,' **** ERROR IN BOT INPUT')
C3680      STOP
C3690      END
C36A0C *****
C36B0C SUBROUTINE CSIMP(F,A,B,DEL,IMAX,S11,S,N,IER)
C36C0C *****
C36D0C COMPLEX SIMPSON INTEGRATION ROUTINE.
C36E0C COMPLEX F,S11,S,SUMK
C36F0C S11=(.0,.0)
C36G0C S=(.C,.0)
C36H0C N=0
C36I0C BA=B-A
C36J0C IF (BA)2C,19,20
C36K0C 19 IER=1
C36L0C RETURN
C36M0C 20 IF(DEL)22,22,23
C36N0C 22 IER=2
C36O0C RETURN
C36P0C 23 IF(IMAX-1)24,24,25
C36Q0C 24 IER=3
C36R0C RETURN
C4010

```

15

```

C4020 25 X=BA/2,+A
C4030 NHALF=1
C4040 SUMK=F(X)*BA*2./3.
C4050 S=SUMK+(F(A)+F(B))*BA/6.
C4060 DO 28 I=2,IMAX
C4070 SI1=S
C4080 S=(S-SUMK/2.)/2.
C4090 NHALF=NHALF*2
C4100 ANHLF=NHALF
C4110 FRSTX=A+(BA/ANHLF)/2.
C4120 SUMK=F(FRSTX)
C4130 XK=FRSTX
C4140 KLAST=NHALF-1
C4150 FINC=BA/ANHLF
C4160 DO 26 K=1,KLAST
C4170 XK=XK+FINC
C4180 26 SUMK=SUMK+F(XK)
C4190 SUMK=SUMK*2.*BA/(3.*ANHLF)
C4200 S=S+SUMK
C4205 IF(I.EQ.2) GO TO 28
C4210 IF(CABS(S).EQ.0.0) GO TO 29
C4220 IF((CABS(S-SI1)/CABS(S))-DEL) 29,28,28
C4230 28 CONTINUE
C4240 IER=4
C4250 WRITE(6,1) SI1,S
C4260 1 FORMAT(/,42H INTEGRATION OF SELF TERM DID NOT CONVERGE,/,
C4270 1 30H THE PREVIOUS AND FINAL VALUES OF THE INTEGRAL ARE,
C4280 2 13H RESPECTIVELY,4E12.3)
C4290 GO TO 30
C4300 29 IER=0
C4310 30 N=2*NHALF
C4320 RETURN
C4330 END
-----
C4340C
C4350 COMPLEX FUNCTION FUNC(E)
C4360C G(NON-SELF TERM) INTEGRAND.
C4370 COMPLEX U
C4380 COMMON /INT/ M,N,BKL,RHO2,DELTA
C4390 DATA U /10.,1./
C4400 R=SQRT(RHO2+E*E)
C4410 FUNC=CEXP(-U*BKL*R)/R*UMN(M,N,E)
C4420 RETURN
C4430 END
-----
C4440C
C4450 COMPLEX FUNCTION FUNCB(E)
C4460C GB(NON-SELF TERM) INTEGRAND.
C4470 COMPLEX U
C4480 COMMON /INT/ M,N,BKL,RHO2,DELTA
C4490 DATA U /10.,1./
C4500 R=SQRT(RHO2+E*E)
C4510 FUNCB=CEXP(-U*BKL*R)/R*UBMN(M,N,E)
C4520 RETURN

```

```

C4530      END
C4540C -----
C4550      FUNCTION UMN(M,N,E)
C4560      COMMON /UCONST/AMN,ABMN,PIN,PIN
C4570      IF(M.NE.N) GO TO 100
C4580      UMN=AMN*(2.-E)*COS(PIN*E)
C4590      RETURN
C4600 100  UMN=AMN*(SIN(PIN*E)-SIN(PIN*E))
C4610      RETURN
C4620      END
C4630C -----
C4640      FUNCTION UMP(M,N,E)
C4650      COMMON /UCONST/AMN,ABMN,PIN,PIN
C4660      IF(M.NE.N) GO TO 100
C4670      UMP=AMN*(-PIN*(2.-E)*SIN(PIN*E)-COS(PIN*E))
C4680      RETURN
C4690 100  UMP=AMN*(PIN*COS(PIN*E)-PIN*COS(PIN*E))
C4700      RETURN
C4710      END
C4720C *****
C4730      COMPLEX FUNCTION FUNCS(E)
C4740C *****
C4750C      G(=SELF TERM) INTEGRAND.
C4760      COMPLEX U
C4770      COMMON /INT/ M,N,BKL,RHO2,DELTA
C4780      DATA PI,U /3.14159265,(0.,1.)/
C4790      IF(E.EQ.0) GO TO 100
C4800      AE=ALOG(E)
C4810      A=ALOG(DELTA+SQRT(DELTA*DELTA+E*E))
C4820      FUNCS=UMN(M,N,E)*(-U*BKL*DELTA+A+U*BKL*E*(A-2.*AE+1.))
C4830      FUNCS=FUNCS+UMP(M,N,E)*E*(AE-1.)
C4840      FUNCS=FUNCS*CEXP(-U*BKL*E)
C4850      RETURN
C4860 100  CONTINUE
C4870      FUNCS=UMN(M,N,0.0)*(-U*BKL*DELTA+ALOG(2.*DELTA))
C4880      RETURN
C4890      END
C4900C *****
C4910      COMPLEX FUNCTION FUNCBS(E)
C4920C *****
C4930C      G(=SELF TERM) INTEGRAND.
C4940      COMPLEX U
C4950      COMMON /INT/ M,N,BKL,RHO2,DELTA
C4960      DATA PI,U /3.14159265,(0.,1.)/
C4970      IF(E.EQ.0) GO TO 100
C4980      AE=ALOG(E)
C4990      A=ALOG(DELTA+SQRT(DELTA*DELTA+E*E))
C5000      FUNCBS=UMN(M,N,E)*(-U*BKL*DELTA+A+U*BKL*E*(A-2.*AE+1.))
C5010      FUNCBS=FUNCBS+UMP(M,N,E)*E*(AE-1.)
C5020      FUNCBS=FUNCBS*CEXP(-U*BKL*E)
C5030      RETURN
C5040 100  CONTINUE

```

```

C5050      FUNCBS=UBMN(M,N,O.C)*(-U*AKL*DELTA+ALOG(2.*DELTA))
C5060      RETURN
C5070      END
C5080C -----
C5090      FUNCTION UBMN(M,N,E)
C5100C NOTE, M,NE.C AND N,NF.O
C5110      COMMON /UCONST/AMN,ABMN,PIM,PIN
C5120      UBMN=ABMN*(2.-E+SIN(PIN*E)/PIN+SIN(PIM*E)/PIM)
C5130      RETURN
C5140      END
C5150C -----
C5160      FUNCTION UBMNP(M,N,E)
C5170C NOTE, M,NE.C AND N,NE.O
C5180      COMMON /UCONST/AMN,ABMN,PIM,PIN
C5190      UBMNP=ABMN*(-1.+COS(PIN*E)+COS(PIM*E))
C5200      RETURN
C5210      END
C5220C *****
C5230      SUBROUTINE PLOT9(X,Y,N,NR)
C5240C *****
C5250C WRITTEN 2/14/74 BY J. M. PUTNAM DEPT 220 X23877
C5260C
C5270C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
C5280C N IS THE NUMBER OF POINTS TO BE PLOTTED.
C5290C NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
C5300C NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
C5310C NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
C5320C
C5330C      REAL X(1),Y(1),HEAD(10)
C5340C      INTEGER LINE(101),BLANK,STAR,PLUS
C5350C      DATA BLANK,STAR,PLUS /1H,1H*,1H+/
C5360C      NC=51
C5370C      N10=(NC-1)/10
C5380C      WRITE(6,500)
C5390C      500 FORMAT(/,44H BODY COORDINATES, + INDICATES TRIANGLE PEAK)
C5400C      WRITE(6,504)
C5410C      XMIN=X(1)
C5420C      XMAX=X(1)
C5430C      YMIN=Y(1)
C5440C      YMAX=Y(1)
C5450C      DO 6 I=1,N
C5460C      IF(X(I).LT.XMIN) XMIN=X(I)
C5470C      IF(X(I).GT.XMAX) XMAX=X(I)
C5480C      IF(Y(I).LT.YMIN) YMIN=Y(I)
C5490C      IF(Y(I).GT.YMAX) YMAX=Y(I)
C5500C      6 CONTINUE
C5510C      DEL=XMAX-XMIN
C5520C      IF(YMAX-YMIN.GT.DEL) DEL=YMAX-YMIN
C5530C      XMAX=XMIN+DEL
C5540C      YMAX=YMIN+DEL
C5550C      DO 5 I=1,N10

```

```

05576      Z=I
05580      5 HEAD(I)=(XMAX-XMIN)*Z/N10+XMIN
05590      DY=(YMAX-YMIN)/(NR-1)
05600      IEDGE=1
05610      IF(X(I).EQ.X(N) .AND. Y(I).EQ.Y(N)) IEDGE=0
05620      Z=YMAX+DY
05630      YL=Z-DY/2.
05640      DO 7 J=1,NR
05650      DO 8 K=1,NC
05660      8 LINE(K)=BLANK
05670      Z=Z-DY
05680      YU=YL
05690      YL=Z-DY/2.
05700      DO 9 I=1,N
05710      IF(Y(I).GE.YU) GO TO 9
05720      IF(Y(I).LT.YL) GO TO 9
05730      K=(X(I)-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
05740      IF(K.GT.NC) K=NC
05750      LINE(K)=STAR
05760      IF(MOD(I,2).EQ.1) LINE(K)=PLUS
05770      IF(IEEDGE.EQ.0) GO TO 9
05780      IF(I.EQ.1 .OR. I.EQ.N) LINE(K)=STAR
05790      9 CONTINUE
05800      WRITE(6,508) Z,(LINE(K),K=1,NC)
05810      7 CONTINUE
05820      WRITE(6,504)
05830      WRITE(6,3002)
05840      WRITE(6,507) XMIN,(HEAD(I),I=1,N10)
05850      RETURN
05860      504 FORMAT(1X,14(1H-),1H.,10(5H----.),1H-)
05870      507 FORMAT(10X,11(F10.4))
05880      508 FORMAT(1X,F12.4,1X,1HI,51A1,1HI)
05890      3002 FORMAT(4X,7HVH/7H,4X,1HI,5(9X,1HI))
05900      END
05910C -----
05920      SUBROUTINE ZLIST(NI,NJ,LI,LJ,NAMI,NAMJ,Z)
05930C WRITE 2 TO THE LINE-PRINTER BY SUBMATRIX BY COLUMNS.
05940      COMPLEX Z(1)
05950      DIMENSION LI(1),LJ(1),NAMI(1),NAMJ(1)
05960      NCOL=C
05970      DO 1CC I=1,NI
05980      1CC NCOL=NCOL+LI(I)
05990      NR=1
06000      DO 3CC J=1,NJ
06010      IF(LJ(J).EQ.0) GO TO 300
06020      NC=1
06030      DO 200 I=1,NI
06040      IF(LI(I).EQ.0) GO TO 200
06050      WRITE(6,1) NAMI(I),NAMJ(J)
06060      1 FORMAT(/,2H Z,1X,A4,3H -,A4)
06070      KI=LI(I)
06080      KJ=LJ(J)

```

```

C609C      K1=(NR-1)*NCOL+NC
C610C      DO 15C KK=1,KJ
C611C      K2=K1+K1-1
C612C      WRITE(6,2) (Z(K),K=K1,K2)
C613C      2    FORMAT(1X,10G11.4)
C614C      15C  K1=K1+NCOL
C615C      NC=NC+11(I)
C616C      20C  CONTINUE
C617C      NR=NR+LJ(J)
C618C      30C  CONTINUE
C619C      RETURN
C620C      END
C621C/EOR

```

PRECEDING PAGE 5

```

C01C0BOTZ,P30,T1CC,CH14,C00,STCX3.
C0110ACCOUNT,MO3265,BOTZ1.
C0120BANNERS(OUTPUT)*J. PUTNAM*DEPT 220*BLD 110-4**
C0130DEFINE(OUTFIL,ZSCX)
C0138FTN(R=0,OPT=2)
C014CLOSET(PRESET=INDEF)
C0150LGO.
C0210EXIT.
C0230/EDR
C0240      PROGRAM BOTZSC(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
C0250      1 OUTFIL,TAPE7=OUTFIL)
C0252C
C0254C BOTZSC GENERATES THE ROT-CAP Z MATRIX.
C0256C
C0258C UNIT 5 IS THE CARD READER.
C0260C UNIT 6 IS THE LINE PRINTER.
C0262C UNIT 7 IS A DISK FILE FOR OUTPUT OF THE Z MATRIX.
C0264C
C0280C      COMPLEX U,A3,Z(1000),G2(3000)
C0281C      COMPLEX Z1,Z2,Z3,Z4
C0282C      COMPLEX Z5,Z6,Z7
C0283C      COMPLEX G1E(1500),G2E(1500)
C0284C      COMMON /WAVE/ BK
C0285C      COMMON /BOT1/ 4*ODE,NPT,N5AND
C0286C      COMMON /BOT2/ NP,BL,YR(83),XR(83),Y91(82),XB1(82)
C0287C      COMMON /BOT3/ O4(82),SV(82),CV(82)
C0288C      COMMON /BOT5/ T(160),TP(160),TZ(160)
C0289C      COMMON /CAP1/ NC,XC,YC,ZC(2)
C0290C      COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
C0291C      COMMON /CAP3/ TCR(36),TCR1(35),TPCR(36),TPCT(36)
C0292C      COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
C0293C      COMMON /EDG1/ NE,ZE(2),ZRE(10)
C0294C      COMMON /EDG2/ TCE(10),TPCE(10),TRE(10),TPBE(10)
C0295C      COMMON /INT/ M,4*O2,ZP
C0296C      DIMENSION LI(2),LJ(3),NAME(2),NAMEJ(3)
C0297C      EXTERNAL FUNC1,FUNC2
C0298C      DATA NAME,NAMEJ /4HS(T),4HS(Z),4HC(T),4HC(R),4HEDGE/
C0299C      U=(0.,1.)
C0300C      PI=3.14159265
C0301C      ETA=376.737
C0302C      READ(5,1) BK
C0303C      FORMAT(E15.7)
C0304C      WRITE(6,2) BK
C0305C      FORMAT(9H1      BK,/,E15.7)
C0306C
C0307C      CALL BOTIN
C0308C
C0309C      CALL CAPIN
C0310C
C0311C      WRITE(7) NP,NC,NPR,NE
C0312C      NM=(NP-3)/2
C0313C      NM2=NM*2

```

```

00500 LR=(NPR-3)/2
00510 LC=NC*NM*LR
00520 LC2=LC*2
00525 LE=NE*NM
00530 LSC=NM2*(LC2+LE)
00540 NP1=NP-1
00550 NPR1=NPR-1
00551 BK1=BK*BL
00552 BK2=BK*BK
00553C CONSTANTS USED IN SUMMATION OF IMPEDANCE ELEMENTS.
00554 C1=BK*ETA
00555 C2=PI*ETA/BK/RL
00556 C3=C1
00557 C4=C2
00558 C3E=ETA/BK
00559 C4E=C1
00560C *** END INITIALIZATION OF PARAMETERS ***
00562C COMPUTATION OF IMPEDANCE MATRICES Z(M), WHERE M = -NMODE+1 TO
00564C NMODE-1.
00566 M=-NMODE
00568 KMODE=2*NMODE-1
00570 DO 998 MM=1,KMODE
00572 M=M+1
00574 WRITE(6,56) M
00576 56 FORMAT(4H1 M=,I3)
00578C COMPUTATION OF GREEN'S FUNCTION KERNEL.
00600 I1=0
00605 NPT2=0
00610 NINT2=0
00615 DO 16 JC=1,NC
00630 DO 16 JP=1,NP1
00640 DO 16 JR=1,NPR1
00645 DO 17 IP=1,NP1
00650 I1=I1+1
00660 YY=YB1(IP)-(YC+RHOC1(JR)*RC1(JP)*SPC(JP))
00670 XX=XB1(IP)-(XC+RHOC1(JR)*RC1(JP)*CPC(JP))
00680 RHQ2=XX*XX+YY*YY
00690 ZP=ZC(JC)
00700 CALL CSIMP(FUNC2,-BL,BL,0.05,10,A3,G2(I1),K,IER)
00710 NPT2=NPT2+K+1
00720 NINT2=NINT2+1
00730 G2(I1)=G2(I1)/4./PI
00740 17 CONTINUE
00750 16 CONTINUE
00760 AVG2=NPT2/FLOAT(NINT2)
00765 WRITE(6,56) NINT2,AVG2
00770 56 FORMAT(/, ' NUMBER OF INTEGRATIONS AND AVERAGE NUMBER OF ',
00775 1 ' PCINTS PER INTEGRATION',/, (I8,G10.3))
00777 IF(ME.EQ.0) GO TO 6C
00780 I1=C
00785 NPT1=0
00790 NINT1=0

```

```

00795      NPT2=C
00800      NINT2=0
00805      DO 18 JC=1,NC
00810      DO 18 JP=1,NP1
00815      DO 18 JZ=1,2
00820      DC 19 IP=1,NP1
00825      I1=I1+1
00830      YY=YB1(IP)-YB1(JP)
00835      XX=XB1(IP)-XB1(JP)
00840      RHC2=XX*XX+YY*YY
00845C CHECK FOR SELF-TERM.
00850      IF (RHC2.EQ.0) RHC2=DH(IP)*DH(IP)/4.
00855      ZP=ZBE((JC-1)*2+JZ)
00860      CALL CSIMP(FUNC2,-BL,BL,0.05,10,A3,G2E(I1),K,IER)
00865      NPT2=NPT2+K+1
00870      NINT2=NINT2+1
00875      G2E(I1)=G2E(I1)/4./PI
00880      IF (MP.NE.1) GO TO 19
00885      CALL CSIMP(FUNC1,-BL,BL,0.05,10,A3,G1E(I1),K,IER)
00890      NPT1=NPT1+K+1
00895      NINT1=NINT1+1
00900      G1E(I1)=G1E(I1)/4./PI
00905      CONTINUE
00910 19 CONTINUE
00915 18 CONTINUE
00920      AVG1=0.0
00925      IF (MP.EQ.1) AVG1=NPT1/FLOAT(NINT1)
00930      AVG2=NPT2/FLOAT(NINT2)
00935 57 WRITE(6,57) NINT1,AVG1,NINT2,AVG2
00940      1 ' NUMBER OF INTEGRATIONS AND AVERAGE NUMBER OF ',
00945      2 ' POINTS PER INTEGRATION (G1 AND G2, RESPECTIVELY)',/,
00950      2 2(I8,G10.3))
00955 60 CONTINUE
00960C *** END CALCULATION OF GREEN'S KERNEL ***
00965C BOT-CAP LOOP.
00970      DO 30 JC=1,NC
00975      DO 30 JP=1,NM
00980      DO 30 JR=1,LR
00985      L1=((JC-1)*NM*LR+(JP-1)*LR+(JR-1))*NM2
00990      L2=L1+NM
00995      L3=L1+NM2*LC
00998      L4=L3+NM
00999      DO 31 IP=1,NM
00999      L1=L1+1
00999      L2=L2+1
00999      L3=L3+1
00999      L4=L4+1
00999      Z1=0.0
00999      Z2=0.0
00999      Z3=0.0
00999      Z4=0.0
00999      JP2=2*(JP-1)
00999      JP7=4*(JP-1)

```

```

C0994      DO 70 JJP=1,4
C0996      JP2=JP2+1
C0998      JP7=JP7+1
C1000      JR2=2*(JR-1)
C1002      JR7=4*(JR-1)
C1004      DO 7C JJR=1,4
C1006      JR2=JR2+1
C1008      JR7=JR7+1
C1010      IP2=2*(IP-1)
C1012      IP7=4*(IP-1)
C1014      DO 71 IIP=1,4
C1016      IP2=IP2+1
C1018      IP7=IP7+1
C1020      AJ=AC(JP2)*ABS(RHOC(JR2+1)**2-RHOC(JR2)**2)
C1022      RJ=RHOC1(JR2)*RC1(JP2)
C1024      J4=((JC-1)*NP1+NPRI+(JP2-1)*NPRI+(JR2-1)*NP1+IP2
C1026      BOT-CAP IMPEDANCE ELEMENTS COMPUTED.
C1028      Z(L1) = ZTT, Z(L2) = ZTT, Z(L3) = ZTR, Z(L4) = ZZR,
C1030      Z1=Z1+DH(IP2)*AJ*((CV(IP2)*CV(JP2)+SV(IP2)*SV(JP2))*
C1032      1 T(IP7)*T(JP7)-TP(IP7)*TP(JP7)/BK/BK/RHOC1(JR2))*
C1034      2 TCT(JR7)*G2(J4)
C1036      Z2=Z2+DH(IP2)*AJ*TZ(IP7)*TP(JP7)*TCT(JR7)/RHOC1(JR2)*
C1038      1 G2(J4)
C1040      Z3=Z3+DH(IP2)*AJ*((CV(IP2)*CPC(JP2)+SV(IP2)*SPC(JP2))*
C1042      1 T(IP7)*TCR(JR7)-TP(IP7)*(TPC(JR7)/RC1(JP2)+TCR(JR7)/RJ)
C1044      2 /BK/BK)*TZ(JP7)*G2(J4)
C1046      Z4=Z4+DH(IP2)*AJ*TZ(IP7)*TZ(JP7)*
C1048      1 (TPC(JR7)/RC1(JP2)+TCR(JR7)/RJ)*G2(J4)
C1050      71 CONTINUE
C1052      7C CONTINUE
C1054      Z(L1)=U*C1*Z1
C1056      Z(L2)=-M*C2*Z2
C1058      Z(L3)=U*C3*Z3
C1060      Z(L4)=-M*C4*Z4
C1062      31 CONTINUE
C1064      30 CONTINUE
C1066      IF(NE.EQ.0) GO TO 900
C1068      BOT-EDGE LOOP.
C1070      DO 130 JC=1,NC
C1072      DO 130 JP=1,NM
C1074      L3=NM2+LC2+((JC-1)*NM+(JP-1))*NM2
C1076      L4=L3+NM
C1078      DO 131 IIP=1,NM
C1080      L3=L3+1
C1082      L4=L4+1
C1084      Z3=0.0
C1086      Z4=0.0
C1088      Z3E=0.0
C1090      Z4E=0.0
C1092      JP2=2*(JP-1)
C1094      JP7=4*(JP-1)
C1096      DO 170 JJP=1,4

```

```

C1610 JP2=JP2+1
C1620 JP7=JP7+1
C1630 JR2=2*LP
C1640 JR7=2*(JC-1)
C1642 JZ2=2*(JC-1)
C1644 JZ7=2*(JC-1)
C1650 DD 170 JJRZ=1,2
C1660 JR2=JR2+1
C1670 JR7=JR7+1
C1672 JZ2=JZ2+1
C1674 JZ7=JZ7+1
C1680 IP2=2*(IP-1)
C1690 IP7=4*(IP-1)
C1700 DD 171 IIP=1,4
C1710 IP2=IP2+1
C1720 IP7=IP7+1
C1730 AJ=AC(JP2)*ABS(RHOC(JR2+1)**2-RHOC(JR2)**2)
C1740 RJ=RHOC1(JR2)*RC1(JP2)
C1750 J4=((JC-1)*NP1*NPRI+(JP2-1)*NPRI+(JR2-1))*NP1+IP2
C1760C BOT-EDGE(CAP) IMPEDANCE ELEMENTS COMPUTED.
C1770C Z3 = ZTR, Z4 = ZZR.
C1830 Z3=Z3+DH(IP2)*AJ*((CV(IP2)*CPC(JP2)+SV(IP2)*SPC(JP2))*
C1840 1 T(IP7)*TCE(JR7)-TP(IP7)*(TPCE(JR7)/RC1(JP2)+TCE(JR7)/RJ)
C1850 2 /BK/BK)*TZ(JP7)*G2(J4)
C1860 Z4=Z4+DH(IP2)*AJ*TZ(IP7)*TZ(JP7)*
C1870 1 (TPCE(JR7)/RC1(JP2)+TCE(JR7)/RJ)*G2(J4)
C1940 J4=((JC-1)*NP1*2+(JP2-1)*2+(JJRZ-1))*NP1+IP2
C1981 AJ=DH(JP2)*ABS(ZE(JC)-ZC(JC))/2.
C1982C BOT-EDGE(BOT) IMPEDANCE ELEMENTS COMPUTED.
C1983C Z(L3) = ZTZ, Z(L4) = ZZZ.
C1984 Z3E=Z3E+DH(IP2)*AJ*TP(IP7)*TZ(JP7)*TPBE(JZ7)*G2E(J4)
C1985 Z4E=Z4E+DH(IP2)*AJ*TZ(IP7)*TZ(JP7)*(TBE(JZ7)*
C1986 1 (G2E(J4)-(-1)**M*G1E(J4))+U*M*PI/BK/BKL*TPBE(JZ7)*G2E(J4))
C1987 171 CONTINUE
C1988 17C CONTINUE
C1991 Z(L3)=U*C3*Z3-U*C3E*Z3E
C1992 Z(L4)=-M*C4*Z4+U*C4E*Z4E
C1993 131 CONTINUE
C1994 130 CONTINUE
C1995C **** END CALCULATION OF ZSC MATRIX ****
C1996 900 CONTINUE
C1998C PRINT OUT IMPEDANCE MATRIX Z(M) BY SUBMATRIX BY COLUMNS,
C2000 LI(1)=NM
C2001 LI(2)=NM
C2002 LJ(1)=LC
C2003 LJ(2)=LC
C2004 LJ(3)=LE
C2005 CALL ZLIST(2,3,LI,LJ,NAMI,NAMJ,Z)
C2012C WRITE ZSC TO DISK.
C2014 WRITE(7) M
C2016 WRITE(7) (Z(I),I=1,LSC)
C2020 998 CONTINUE

```

```

C2036 STOP
C2040 END
C2050C -----
C2055 SUBROUTINE BOTIN
C2060C READ BOT COORDINATES AND COMPUTE BOT SEGMENT ARRAYS.
C2070C
C2075 COMMON /BOT1/ NMODE,NPT,NBAND
C2080 COMMON /BOT2/ NP,RL,YR(83),YB(82),XB(82)
C2085 COMMON /BOT3/ DH(82),SV(82),CV(82)
C2090 COMMON /BOT5/ T(160),TP(150),TZ(160)
C2095 COMMON /BOT6/ IEDGE,IUNIF
C2100 READ(5,49) NMODE,NPT,NBAND
C2105 READ(5,49) NP
C2110 49 FORMAT(3I3)
C2115 WRITE(6,45) NMODE,NPT,NBAND,NP
C2120 48 FORMAT(32I4,NMODE,NPT,NBAND,NP,/,4I8)
C2125 READ(5,53)(YB(I),I=1,NP)
C2130 READ(5,53)(XB(I),I=1,NP)
C2135 53 FORMAT(1CF8.4)
C2140 WRITE(6,55)
C2145 55 FORMAT(/,3H YB)
C2150 WRITE(6,46)(YB(I),I=1,NP)
C2155 46 FORMAT(1X,10F8.4)
C2160 WRITE(6,56)
C2165 56 FORMAT(/,3H XB)
C2170 WRITE(6,46)(XB(I),I=1,NP)
C2175C PLOT THE BODY COORDINATES.
C2180 CALL PLOTB(XB,YB,NP,41)
C2185 READ(5,53) BL
C2190 WRITE(6,47) BL
C2195 47 FORMAT(/,21H HALF-LENGTH OF BOT =,F12.4)
C2200C CHECK IF BOT CONTAINS AN ODD NUMBER OF POINTS.
C2205 IF(MOD(NP,2).NE.1) GO TO 980
C2210C CHECK IF GENERATING CURVE IS OPEN OR CLOSED.
C2215 IEDGE=1
C2220 IF((YB(1)-YB(NP)).NE.0..OR.(XB(1)-XB(NP)).NE.0.) GO TO 10
C2225 IEDGE=0
C2230 YB(NP+1)=YB(2)
C2235 XB(NP+1)=XB(2)
C2240 YB(NP+2)=YB(3)
C2245 XB(NP+2)=XB(3)
C2250 NP=NP+2
C2255 WRITE(6,66) NP
C2260 66 FORMAT(/,30H BOT GENERATING CURVE IS CLOSED. NP = ,I3)
C2265 10 CONTINUE
C2270C COMPUTATION OF BODY SEGMENT PARAMETERS.
C2275 DO 57 I=2,NP
C2280 I2=I-1
C2285 RR1=YB(I)-YB(I2)
C2290 RR2=XB(I)-XB(I2)
C2295 DH(I2)=SQRT(RR1*RR1+RR2*RR2)

```

```

02300 XB1(I2)=.5*(XB(I)+XB(I2))
02305 YB1(I2)=.5*(YB(I)+YB(I2))
02310 SV(I2)=RR1/DH(I2)
02315 CV(I2)=RR2/DH(I2)
02320 57 CONTINUE
02325C CHECK IF BOT SEGMENTATION IS UNIFORM.
02330 IUNIF=0
02335 NP1=NP-1
02340 DO 60 I=2,NP1
02345 RR1=DH(I)/DH(1)
02350 IF(RR1.LT.0.99 .OR. RR1.GT.1.01) GO TO 20
02355 60 CONTINUE
02360 IUNIF=1
02365 WRITE(6,67)
02370 67 FORMAT(//,' BOT GENERATING CURVE HAS UNIFORM SEGMENTATION')
02375 20 CONTINUE
02380C COMPUTATION OF TRIANGLE FUNCTIONS T.
02385 NM=(NP-3)/2
02390 DO 74 J=1,NM
02395 J2=2*(J-1)+1
02400 J3=J2+1
02405 J4=J3+1
02410 J5=J4+1
02415 J6=4*(J-1)+1
02420 J7=J6+1
02425 J8=J7+1
02430 J9=J8+1
02435 DEL1=DH(J2)+DH(J3)
02440 DEL2=DH(J4)+DH(J5)
02445 TP(J6)=1./DEL1
02450 TP(J7)=1./DEL1
02455 TP(J8)=1./DEL2
02460 TP(J9)=1./DEL2
02465 T(J6)=DH(J2)/2./DEL1
02470 T(J7)=(DH(J2)+DH(J3)/2.)/DEL1
02475 T(J8)=(DH(J4)/2.+DH(J5))/DEL2
02480 T(J9)=DH(J5)/2./DEL2
02485 74 CONTINUE
02490 NM4=NM+4
02495 DO 75 J=1,NM4
02500 75 TZ(J)=T(J)
02505 IF(IEEDGE.EQ.0) GO TO 76
02510 TZ(1)=2.-T(1)
02515 TZ(2)=2.-T(2)
02520 TZ(NM4-1)=2.-T(NM4-1)
02525 TZ(NM4)=2.-T(NM4)
02530 76 CONTINUE
02535 RETURN
02540 98C WRITE(6,981)
02545 981 FORMAT(//,' **** ERROR IN BOT INPUT')
02550 STOP
02555 END

```

```

C3090C *****
C3100C SUBROUTINE PLOTR(X,Y,N,NP)
C3110C *****
C3120C
C3130C WRITTEN 2/14/74 BY J. M. PUTNAM DEPT 220 X23877
C3140C
C3150C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
C3160C N IS THE NUMBER OF POINTS TO BE PLOTTED.
C3170C NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
C3180C NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
C3190C NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
C3200C
C3210C REAL X(1),Y(1),HEAD(10)
C3220C INTEGER LINE(101),BLANK,STAR,PLUS
C3230C DATA BLANK,STAR,PLUS /1H,1H*,1H+ /
C3240C NC=51
C3250C N10=(NC-1)/10
C3260C WRITE(6,500)
C3270C 500 FORMAT(//,44H BODY COORDINATES, + INDICATES TRIANGLE PEAK)
C3280C WRITE(6,504)
C3290C XMIN=X(1)
C3300C XMAX=X(1)
C3310C YMIN=Y(1)
C3320C YMAX=Y(1)
C3330C DO 6 I=1,N
C3340C IF(X(I).LT.XMIN) XMIN=X(I)
C3350C IF(X(I).GT.XMAX) XMAX=X(I)
C3360C IF(Y(I).LT.YMIN) YMIN=Y(I)
C3370C IF(Y(I).GT.YMAX) YMAX=Y(I)
C3380C 6 CONTINUE
C3390C DEL=XMAX-XMIN
C3400C IF(YMAX-YMIN.GT.DEL) DEL=YMAX-YMIN
C3410C XMAX=XMIN+DEL
C3420C YMAX=YMIN+DEL
C3430C DO 5 I=1,N10
C3440C Z=I
C3450C 5 HEAD(I)=(XMAX-XMIN)*Z/N10+XMIN
C3460C DY=(YMAX-YMIN)/(NR-1)
C3470C IEDGE=1
C3480C IF(X(1).EQ.X(N) .AND. Y(1).EQ.Y(N)) IEDGE=0
C3490C Z=YMAX+DY
C3500C YL=Z-DY/2.
C3510C DO 7 J=1,NR
C3520C DO 8 K=1,NC
C3530C 8 LINE(K)=BLANK
C3540C Z=Z-DY
C3550C YU=YL
C3560C YL=Z-DY/2.
C3570C DO 9 I=1,N
C3580C IF(Y(I).GE.YU) GO TO 9
C3590C IF(Y(I).LT.YL) GO TO 9
C3600C K=(X(I)-XMIN)/(XMAX-XMIN)*(NC-1)+1.5

```

```

C361C      IF(K.GT.NC) K=NC
C362C      LINE(K)=STAR
C363C      IF(MOD(I,2).EQ.1) LINE(K)=PLUS
C364C      IF(IEDGE.EQ.0) GO TO 9
C365C      IF(I.EQ.1 .OR. I.EQ.N) LINE(K)=STAR
C366C  9     CONTINUE
C367C      WRITE(6,508) Z,(LINE(K),K=1,NC)
C368C  7     CONTINUE
C369C      WRITE(6,504)
C370C      WRITE(6,506)
C371C      WRITE(6,507) XMIN,(HEAD(I),I=1,N10)
C372C      RETURN
C373C  5C4    FORMAT (1X,14(1H-),1H.,10(5H----.),1H-)
C374C  507    FORMAT(1X,11(F10.4))
C375C  508    FORMAT(1X,F12.4,1X,1HI,51A1,1HI)
C376C  3002   FORMAT(4X,7HYH / XH,4X,1HI,5(9X,1HI))
C377C      END
-----
C3780C      SUBROUTINE CAPIN
C3785
C3790C      READ CAP INPUTS AND COMPUTE CAP ARRAYS.
C3800C
C3805C      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
C3810C      COMMON /CAP1/ NC,XC,YC,ZC(2)
C3815C      COMMON /CAP2/ NPR,RHOC(21),PHOC1(20),DRHOC(20)
C3820C      COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
C3825C      COMMON /CAP4/ RC(93),RC1(92),AC(92),CPC(82),SPC(82)
C3830C      COMMON /EDG1/ NE,ZE(2),ZBE(10)
C3835C      COMMON /EDG2/ TCE(10),TPCE(10),TBE(10),TPBE(10)
C3840C      READ(5,1) NC,NPR,NE
C3845C  1     FORMAT(3I3)
C3850C      IF(NE.NE.0) NE=NC
C3855C      WRITE(6,3) NC,NPR,NE
C3860C  3     FORMAT(16H1 NC NPR NE,/,3I5,/)
C3865C      IF(NC.EQ.0) RETURN
C3870C      READ(5,2) XC,YC
C3875C  2     FORMAT(10F8.4)
C3880C      READ(5,2) (ZC(I),I=1,NC)
C3885C      READ(5,2) (RHOC(I),I=1,NPR)
C3890C      IF(NE.NE.0) READ(5,2) (ZE(I),I=1,NE)
C3895C      WRITE(6,4)
C3900C  4     FORMAT(37H CAP XC YC ZC ZE,/)
C3905C      DO 1CC I=1,NC
C3910C      IF(NE.EQ.0) WRITE(6,5) I,XC,YC,ZC(I)
C3915C      IF(NE.NE.0) WRITE(6,5) I,XC,YC,ZC(I),ZE(I)
C3920C  10C    CONTINUE
C3925C  5     FORMAT(14,4X,4F8.4)
C3930C      WRITE(6,6)
C3935C  6     FORMAT(7,5H RHOC)
C3940C      WRITE(6,7) (RHOC(I),I=1,NPR)
C3945C  7     FORMAT(1X,10F8.4)
C3950C      IF(MOD(NPR,2).NE.1) GO TO 990

```

```

C3955      DO 120 I=2,NPR
C3960 120  IF(RHCC(I).LE,RHOC(I-1)) GO TO 980
C3965C COMPUTATION OF CAP SECTOR PARAMETERS.
C3970      DO 47 I=1,NP
C3975      RC(I)=SQRT((YB(I)-YC)**2+(XB(I)-XC)**2)
C3980 47  CONTINUE
C3985      DO 57 I=2,NP
C3990      I2=I-1
C3995      RR1=YB1(I2)-YC
C4000      RR2=XB1(I2)-XC
C4005      RC1(I2)=SQRT(RR1**2+RR2**2)
C4010      AC(I2)=ABS((XR(I)-XC)*(YB(I)+YC)+(XB(I2)-XB(I))*
C4015 1  (YB(I2)+YB(I))+(XC-XR(I2))*(YC+YB(I2)))/2.
C4020      SPC(I2)=RR1/RC1(I2)
C4025      CPC(I2)=RR2/RC1(I2)
C4030 57  CONTINUE
C4035      DO 67 I=2,NPR
C4040      I2=I-1
C4045      RHOC(I2)=(RHOC(I)+RHOC(I2))/2.
C4050      DRHOC(I2)=RHOC(I)-RHOC(I2)
C4055 67  CONTINUE
C4060C COMPUTATION OF CAP TRIANGLE FUNCTIONS.
C4065      LC=(NPR-3)/2
C4070      DO 74 J=1,LC
C4075      J2=2+(J-1)*1
C4080      J3=J2+1
C4085      J4=J3+1
C4090      J5=J4+1
C4095      J6=4+(J-1)*1
C4100      J7=J6+1
C4105      J8=J7+1
C4110      J9=J8+1
C4115      DEL1=DRHOC(J2)+DRHOC(J3)
C4120      DEL2=DRHOC(J4)+DRHOC(J5)
C4125      TPCR(J6)=1./DEL1
C4130      TPCR(J7)=1./DEL1
C4135      TPCR(J8)=-1./DEL2
C4140      TPCR(J9)=-1./DEL2
C4145      TCR(J6)=DRHOC(J2)/2./DEL1
C4150      TCR(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
C4155      TCR(J8)=(DRHOC(J4)/2.+DRHOC(J5))/DEL2
C4160      TCR(J9)=DRHOC(J5)/2./DEL2
C4165 74  CONTINUE
C4170      LC4=LC*4
C4175      DO 75 I=1,LC4
C4180      TCT(I)=TCR(I)
C4185      TPCT(I)=TPCR(I)
C4190 75  CONTINUE
C4195      TCT(LC4-1)=2.-TCT(LC4-1)
C4200      TCT(LC4)=2.-TCT(LC4)
C4205      TPCT(LC4-1)=-TPCT(LC4-1)
C4210      TPCT(LC4)=-TPCT(LC4)

```

```

C4215 IF(RHOC(1).EQ.0.0) GO TO 76
C4220 TCT(1)=2.-TCT(1)
C4225 TCT(2)=2.-TCT(2)
C4230 TPCT(1)=-TPCT(1)
C4235 TPCT(2)=-TPCT(2)
C4240 76 CONTINUE
C4245 IF(NE.EQ.0) RETURN
C4250C COMPUTATION OF EDGE HALF TRIANGLE FUNCTIONS.
C4255 DO 8C IC=1,NC
C4260 J2=NPR-2
C4265 J3=J2+1
C4270 J6=2*(IC-1)+1
C4275 J7=J6+1
C4280 DEL1=DRHOC(J2)+DRHOC(J3)
C4285 TPCF(J6)=1./DEL1
C4290 TPCF(J7)=1./DEL1
C4295 TCF(J6)=DRHOC(J2)/2./DEL1
C4300 TCF(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
C4305 DEL2=ZE(IC)-ZC(IC)
C4310 ZBFE(J6)=ZC(IC)+0.25*DEL2
C4315 ZBFE(J7)=ZC(IC)+0.75*DEL2
C4320 IF(DEL2.LT.0.0) GO TO 78
C4325C EDGE IS AT Z=-L
C4330 TPCF(J6)=1./DEL2
C4335 TPCF(J7)=1./DEL2
C4340 TBF(J6)=0.75
C4345 TBF(J7)=0.25
C4350 GO TO 80
C4355 78 CONTINUE
C4360C EDGE IS AT Z=L
C4365 TPCF(J6)=1./DEL2
C4370 TPCF(J7)=1./DEL2
C4375 TBF(J6)=0.75
C4380 TBF(J7)=0.25
C4385 8C CONTINUE
C4390 RETURN
C4395 980 WRITE(6,981)
C4400 981 FORMAT(//,1) **** ERROR IN CAP INPUT!
C4405 STOP
C4410 END
C4415C *****
C4420C SUBROUTINE CSIMP(F,A,B,DEL,IMAX,SII,S,N,IER)
C4425C *****
C4430C COMPLEX SIMPSON INTEGRATION ROUTINE.
C4435C COMPLEX F,SII,S,SUMK
C4440C SII=(.C.,0)
C4445C S=(.C.,.0)
C4450C N=0
C4455C BA=B-A
C4460C IF (BA)20,19,20
C4465C 19 IER=1
C4470C RETURN

```

```

C5030 20 IF (DEL) 22, 22, 23
C5040 22 IER=2
C5050 RETURN
C5060 23 IF (IMAX-1) 24, 24, 25
C5070 24 IER=3
C5080 RETURN
C5090 25 X=BA/2, +A
C5100 NHALF=1
C5110 SUMK=F(X)*BA*2./3.
C5120 S=SUMK+(F(A)+F(B))*BA/6.
C5130 DO 28 I=2, IMAX
C5140 SI1=S
C5150 S=(S-SUMK/2.)/2.
C5160 NHALF=NHALF*2
C5170 ANHLF=NHALF
C5180 FRSTX=A+(BA/ANHLF)/2.
C5190 SUMK=F(FRSTX)
C5200 XK=FRSTX
C5210 KLAST=NHALF-1
C5220 FINC=BA/ANHLF
C5230 DO 26 K=1, KLAST
C5240 XK=XK+FINC
C5250 26 SUMK=SUMK+F(XK)
C5260 SUMK=SUMK+2.*BA/(3.*ANHLF)
C5270 S=S+SUMK
C5280 IF (CABS(S).EQ.0.0) GO TO 29
C5290 IF ((CABS(S-SI1)/CABS(S))-DEL) 29, 28, 28
C5300 28 CONTINUE
C5310 IER=4
C5320 WRITE(6,1) SI1,S
C5330 1 FORMAT(//,29H INTEGRATION DID NOT CONVERGE,//
C5340 1 50H THE PREVIOUS AND FINAL VALUES OF THE INTEGRAL ARE,
C5350 2 13H RESPECTIVELY,4E12.3)
C5360 GO TO 30
C5370 29 IER=C
C5380 30 N=2*NHALF
C5390 RETURN
C5400 END
C5410C *****
C5412C COMPLEX FUNCTION FUNC1(E)
C5414C *****
C5416C COMPLEX U
C5418C COMMON /WAVE/ AK
C5420C COMMON /INT/ M,RHO2,7P
C5422C DATA PI,L /3.14159265,(0.,1.)/
C5424C R=SQRT(RHO2+(E-7P)**2)
C5426C FUNC1=CEXP(-U*AK*R)/R
C5428C RETURN
C5430C END
C5432C *****
C5434C COMPLEX FUNCTION FUNC2(E)
C5436C *****

```

```

05438      COMPLEX U
05440      COMMON /WAVE/ BK
05442      COMMON /BOTZ/ NP,PL
05444      COMMON /INT/ M,RH02,ZP
05446      DATA PI,U /3.14159265,(0.,1.)/
05448      R=SQRT(RH02+(E-ZP)**2)
05450      FUNC2=CEXP(-U*(M*PI*E/BL+BK*R))/R
05452      RETURN
05454      END
-----
05460C      SUBROUTINE ZLIST(NI,NJ,LI,LJ,NAMI,NAMJ,Z)
05464      WRITE Z TO THE LINE-PRINTER BY SUBMATRIX BY COLUMNS.
05474C      COMPLEX Z(1)
05484      DIMENSION LI(1),LJ(1),NAMI(1),NAMJ(1)
05494      NCCL=C
05504      DO 1CC I=1,NI
05514      1CC      NCOL=NCOL+LI(I)
05524      NR=1
05534      DO 3CC J=1,NJ
05544      IF(LJ(J).EQ.0) GO TO 300
05554      NC=1
05564      DO 2CC I=1,NI
05574      IF(LI(I).EQ.0) GO TO 200
05584      WRITE(6,1) NAMI(I),NAMJ(J)
05594      1      FORMAT(1,2H Z,1X,A4,3H - ,A4)
05604      KI=LI(I)
05614      KJ=LJ(J)
05624      K1=(NR-1)*NCOL+NC
05634      DO 15C KK=1,KJ
05644      K2=K1+KI-1
05654      WRITE(6,2) (Z(K),K=K1,K2)
05664      2      FORMAT(1X,10G11.4)
05674      K1=K1+NCOL
05684      15C      NC=NC+LI(I)
05694      2CC      CONTINUE
05704      NR=NR+LJ(J)
05714      30C      CONTINUE
05724      RETURN
05734      END
05744      END
05754/ECR

```



```

00100BOTZ,P30,T1C0,CH14,000,STCX3.
00110ACCOUNT,NO3265,BOTZ2A.
00120BANNERS(OUTPUT)*J,BUTNAM*DEPT 220*BLO 110-4**
00130DEFINE(OUTFIL=ZCCX)
00138FTN(R=0,OPT=2)
00140LDSET(PRESET=INDEF)
00150LGO.
00210EXIT.
00230/EOR
00240      PROGRAM BOTZCC(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
00250      1 OUTFIL,TAPE7=OUTFIL)
00252C
00254C BOTZCC GENERATES THE CAP Z MATRIX.
00256C
00258C UNIT 5 IS THE CARD READER.
00260C UNIT 6 IS THE LINE POINTER.
00262C UNIT 7 IS A DISK FILE FOR OUTPUT OF THE Z MATRIX.
00264C
00280      COMPLEX U,A3,Z(9216),GS(1640)
00281      DIMENSION TNGHT(36),RWGHT(360),ENGHT(40)
00282      COMPLEX Z1,Z2,Z3,Z4
00283      COMPLEX Z3E,Z4E
00284      COMMON /WAVE/ BK
00285      COMMON /BOT1/ NMJDE,NPT,NBAND
00287      COMMON /BOT2/ NP,RL,YR(83),XR(83),Y91(82),XB1(82)
00288      COMMON /BOT3/ DH(82),SV(82),CV(82)
00289      COMMON /BOT5/ T(160),TP(160),TZ(160)
00290      COMMON /CAP1/ NC,XC,YC,ZC(2)
00292      COMMON /CAP2/ 4PR,RHOC(21),RHOC1(20),DRHOC(20)
00294      COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
00296      COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
00298      COMMON /EDG1/ NE,ZE(2),ZRE(10)
00300      COMMON /EDG2/ TCE(10),TPCE(10),TBE(10),TPBE(10)
00302      COMMON /INT/ CONST1,CONST2
00303      DIMENSION LI(3),LJ(3),NAMZ(3),NAMJ(3)
00304      EXTERNAL FUNC
00305      DATA NAM1,NAMJ /4HC(T),4HC(R),4HEDGE,4HC(T),4HC(R),4HEDGE/
00306      U=(0.,1.)
00307      PI=3.14159265
00308      ETA=376.707
00309      READ(5,1) BK
00310      1 FORMAT(E15.7)
00311      WRITE(6,2) BK
00312      2 FORMAT(9H1      BK,/,E15.7)
00313
00314      CALL BOTIN
00315
00316      CALL CAPIN
00317
00318      WRITE(7) NC,NPR,NE
00319      NM=(NP-3)/2
00320      LR=(NPR-3)/2

```

```

00510      LC=NC*NM*LR
00520      LC2=LC*2
00525      LE=NE*NM
00530      LCC=(LC2+LE)**2
00540      NP1=NP-1
00550      NPR1=NPR-1
00552C    CALCULATE IMPEDANCE LOADING WEIGHTS ON CAP.
00554      J=0
00556      DO 20 JP=1,NM
00558      J2=2*(JP-1)+1
00560      J3=J2+1
00562      J4=J3+1
00564      J5=J4+1
00566      J6=4*(JP-1)+1
00568      J7=J6+1
00570      J8=J7+1
00572      J9=J8+1
00574      DO 10 JR=1,LR
00576      J=J+1
00578      J1=2*(JR-1)+1
00580      TWGHT(J)=(RHOC(J1+4)**2-RHOC(J1)**2)*
00582      1 (AC(J2)*T(J6)+AC(J3)*T(J7)+AC(J4)*T(J8)+AC(J5)*T(J9))
00584      RWGHT(J)=(RHOC(J1+4)**2-RHOC(J1)**2)*(AC(J2)*TZ(J6)+
00586      1 AC(J3)*TZ(J7)+AC(J4)*TZ(J8)+AC(J5)*TZ(J9))
00588      10 CONTINUE
00590      IF(NE.EQ.0) GO TO 20
00592      EWGHT(JP)=(RHOC(J1+4)**2-RHOC(J1+2)**2)*(AC(J2)*TZ(J6)+
00594      1 AC(J3)*TZ(J7)+AC(J4)*TZ(J8)+AC(J5)*TZ(J9))+
00596      2 ABS(ZE(I)-ZC(I))*(DH(J2)*TZ(J6)+DH(J3)*TZ(J7)+
00598      3 DH(J4)*TZ(J8)+DH(J5)*TZ(J9))
00600      20 CONTINUE
00602      I1=NM*LR
00604      WRITE(7) (TWGHT(I),I=1,I1)
00606      WRITE(7) (RWGHT(I),I=1,I1)
00608      IF(NE.NE.0) WRITE(7) (EWGHT(I),I=1,NM)
00610C    COMPUTATION OF GREEN'S FUNCTION KERNEL SELF TERM.
00740      I1=C
00742      NPT1=0
00744      NINT1=0
00746      DO 17 JP=1,NP1
00748      DO 17 JR=1,NPR1
00750      I1=I1+1
00770C    SELF TERM.
00790      R1=RC1(JP)*RHOC(JR)
00800      R2=RC1(JP)*RHOC(JR+1)
00810      C2TH1=-(DH(JP)**2-RC(JP+1)**2-RC(JP)**2)/2./
00820      1 RC(JP+1)/RC(JP)
00830      TH1=ACOS(C2TH1)/2.
00840      CONST1=(R1+R2)**2/4.
00850      CONST2=-(R1+R2)*COS(TH1)
00860      CALL CSIMP(FUNC,R1,R2,0.05,10,A3,GS(I1),K,IER)
00870      NPT1=NPT1+K+1

```

```

C0880      NINT1=NINT1+1
C0890      GS(I1)=GS(I1)*(R1+R2)*(R2-R1)*TH1*TH1/2./PI
C0900 17    CONTINUE
C0910      AVG1=NPT1/FLOAT(NINT1)
C0920      WRITE(6,56) NINT1,AVG1
C0930 56    FORMAT(7,' NUMBER OF INTEGRATIONS AND AVERAGE NUMBER OF ',
C0940      1,' POINTS PER INTEGRATION FOR SELF TERM',/,(I8,G10.3))
C0950C *** END CALCULATION OF GREEN'S KERNEL ***
C0951C CAP-CAP LOOP.
C0952      DO 30 JC=1,NC
C0954      DO 30 JP=1,NH
C0956      DO 30 JR=1,LR
C0958      L1=((JC-1)*NH*LR+(JP-1)*LR+(JR-1))*(LC2+LE)
C0960      L2=L1+LC
C0962      L3=L1+LC*(LC2+LE)
C0964      L4=L3+LC
C0966      DO 31 IC=1,NC
C0968      DO 31 IP=1,NH
C0970      DO 31 IR=1,LR
C0972      L1=L1+1
C0974      L2=L2+1
C0976      L3=L3+1
C0978      L4=L4+1
C0980      Z1=0.0
C0982      Z2=0.0
C0984      Z3=0.0
C0986      Z4=0.0
C0990      JP2=2*(JP-1)
C0992      JP7=4*(JP-1)
C0994      DO 70 JJP=1,4
C0996      JP2=JP2+1
C0998      JP7=JP7+1
C1000      JR2=2*(JR-1)
C1002      JR7=4*(JR-1)
C1004      DO 70 JJR=1,4
C1006      JR2=JR2+1
C1008      JR7=JR7+1
C1010      IP2=2*(IP-1)
C1012      IP7=4*(IP-1)
C1014      DO 71 IIP=1,4
C1016      IP2=IP2+1
C1018      IP7=IP7+1
C1020      IR2=2*(IR-1)
C1022      IR7=4*(IR-1)
C1024      DO 71 IIR=1,4
C1026      IR2=IR2+1
C1028      IR7=IR7+1
C1030      IF(IC.NE.JC) GO TO 60
C1032      IF(IR2.NE.JR2) GO TO 60
C1034      IF(IP2.NE.Y91(JP2)) GO TO 60
C1036      IF(X81(IP2).NE.X81(JP2)) GO TO 60
C1038C SELF TERM.

```

```

C1035      A3=GS(IR2+(IP2-1)*NPR1)
C1036      GO TO 65
C1037C NON-SELF TERM.
C1038      60 CONTINUE
C1042      XX=RHOC1(JR2)*RC1(JP2)*CPC(JP2)-RHOC1(IR2)*RC1(IP2)*CPC(IP2)
C1052      YY=RHOC1(JR2)*RC1(JP2)*SPC(JP2)-RHOC1(IR2)*RC1(IP2)*SPC(IP2)
C1062      ZZ=ZC(JC)-ZC(IC)
C1072      RH02=XX*XX+YY*YY+ZZ*ZZ
C1102      AJ=AC(JP2)*ABS(RHOC(JR2+1)**2-RHOC(JR2)**2)
C1112      AI=AC(IP2)*ABS(RHOC(IR2+1)**2-RHOC(IR2)**2)
C1122      RR=SGRT(RH02)
C1132      A3=AI*AJ*CEXP(-U*BK*RR)/4./PI/RR
C1142      65 CONTINUE
C1152      RI=RHOC1(IR2)*RC1(IP2)
C1162      RJ=RHOC1(JR2)*RC1(JP2)
C1172C CAP-CAP IMPEDANCE ELEMENTS COMPUTED.
C1202C Z(L1) = ZTT , Z(L2) = ZRT , Z(L3) = ZTR , Z(L4) = ZRR.
C1212      Z1=Z1+TCT(IR7)*TCT(JR7)*((CV(IP2)*CV(JP2)+
C1222      1 SV(IP2)*SV(JP2))*T(IP7)*T(JP7)
C1232      2 -T(IP7)*T(JP7)/RK/BK/RHOC1(IR2)/RHOC1(JR2))*A3
C1242      Z2=Z2+TZ(IP7)*TCT(JR7)*((CPC(IP2)*CV(JP2)+
C1252      1 SPC(IP2)*SV(JP2))*TCR(IR7)*T(JP7)
C1262      2 -T(IP7)*TCR(IR7)/RC1(IP2)*TCR(IR7)/RJ)*
C1272      3 T(JP7)/RHOC1(JR2)/BK/BK)*A3
C1282      Z3=Z3+TCT(IR7)*TZ(JP7)*((CV(IP2)*CPC(JP2)+
C1292      1 SV(IP2)*SPC(JP2))*T(IP7)*TCR(JR7)
C1302      2 -T(IP7)/RHOC1(JR2)*
C1312      3 (TPCR(JR7)/RC1(JP2)+TCR(JR7)/RJ)/RK/BK)*A3
C1322      Z4=Z4+TZ(IP7)*TZ(JP7)*((CPC(IP2)*CPC(JP2)+
C1332      1 SPC(IP2)*SPC(JP2))*TCR(IR7)*TCR(JR7)
C1342      2 -T(IP7)*TCR(IR7)/RC1(IP2)*TCR(IR7)/RJ)*
C1352      3 (TPCR(JR7)/RC1(JP2)+TCR(JR7)/RJ)/BK/BK)*A3
C1362      71 CONTINUE
C1372      70 CONTINUE
C1382      Z(L1)=U*BK*ET*Z1
C1392      Z(L2)=U*BK*ETA*Z2
C1402      Z(L3)=U*BK*ETA*Z3
C1412      Z(L4)=U*BK*ETA*Z4
C1422      31 CONTINUE
C1432      30 CONTINUE
C1442C IF(NE.EQ.0) GO TO 900
C1452C CAP-FOGE LOOP.
C1462      DO 130 JC=1,NC
C1472      DO 130 JP=1,NM
C1482      L3=(LC2+LE)*LC2+((JC-1)*NM+(JP-1))*(LC2+LE)
C1492      L4=L3+LC
C1502      DO 131 IC=1,NC
C1512      DO 131 IP=1,NM
C1522      DO 131 IR=1,LR
C1532      L3=L3+1
C1542      L4=L4+1
C1552      Z3=0.0

```

```

01582      Z4=0.0
01584      Z3E=0.0
01586      Z4E=0.0
01588      JP2=2*(JP-1)
01590      JP7=4*(JP-1)
01592      DO 170 JJP=1,4
01594      JP2=JP2+1
01596      JP7=JP7+1
01598      JR2=2*LR
01600      JR7=2*(JC-1)
01602      JZ2=2*(JC-1)
01604      JZ7=2*(JC-1)
01606      DO 170 JJRZ=1,2
01608      JR2=JR2+1
01610      JR7=JR7+1
01612      JZ2=JZ2+1
01614      JZ7=JZ7+1
01616      IP2=2*(IP-1)
01618      IP7=4*(IP-1)
01620      DO 171 IIP=1,4
01622      IP2=IP2+1
01624      IP7=IP7+1
01626      IR2=2*(IR-1)
01628      IR7=4*(IR-1)
01630      DO 171 IIR=1,4
01632      IR2=IR2+1
01634      IR7=IR7+1
01636      AIC=AC(IP2)*ABS(RHOC(IR2+1)**2-RHOC(IR2)**2)
01638      AJC=AC(JP2)*ABS(RHOC(JR2+1)**2-RHOC(JR2)**2)
01640      AJB=CH(JP2)*ABS(ZE(JC)-ZC(JC))/2.
01642      IF(IC.NE.JC) GO TO 160
01644      IF(JR2.NE.JR2) GO TO 160
01646      IF(YB1(IP2).NE.YB1(JP2)) GO TO 160
01648      IF(XB1(IP2).NE.XB1(JP2)) GO TO 160
01650      SELF TERM
01652      A3=63*(IR2+(IP2-1)*NPR1)
01654      GO TO 165
01656      NON-SELF TERM.
01658      160 CONTINUE
01660      XX=RHOC1(JR2)*RC1(JP2)*CPC(JP2)-RHOC1(IR2)*RC1(IP2)*CPC(IP2)
01662      YY=RHOC1(JR2)*RC1(JP2)*SPC(JP2)-RHOC1(IR2)*RC1(IP2)*SPC(IP2)
01664      ZZ=ZC(JC)-ZC(IC)
01666      RHO2=XX*XX+YY*YY+ZZ*ZZ
01668      RR=SQRT(RHO2)
01670      A3=AIC*AJC*CEXP(-U*BK*RR)/4./PI/RR
01672      165 CONTINUE
01674      RI=RHOC1(IR2)*RC1(IP2)
01676      RJ=RHOC1(JR2)*RC1(JP2)
01678      CAP-EDGE(CAP) IMPEDANCE ELEMENTS COMPUTED.
02002C      Z(L3) = ZTR, Z(L4) = ZRR
02004C      Z3=Z3+TCT(IP7)*TZ(JP7)*((CV(IP2)*CPC(JP2)+
02006C      1 SV(IP2)*SPC(JP2))*T(IP7)*TCE(JR7)

```

```

C2102      2 -TP(IP7)/RHOC1(IR2)*
C2112      3 (TPCE(JR7)/RC1(JP2)+TCE(JR7)/RJ)/BK/BK)*A3
C2122      Z4=Z4+TZ(IP7)*TZ(JP7)*((CPC(IP2)*CPC(JP2)+
C2132      1 SPC(IP2)+SPC(JP2))*TCR(IR7)+TCE(JR7)
C2142      2 -(TPCR(IR7)/RC1(IP2)+TCR(IR7)/RI)*
C2152      3 (TPCE(JR7)/RC1(JP2)+TCE(JR7)/RJ)/BK/BK)*A3
C2162      XX=XB1(JP2)-(XC+RHOC1(IR2)*RC1(IP2)*CPC(IP2))
C2172      YY=YB1(JP2)-(YC+RHOC1(IR2)*RC1(IP2)*CPC(IP2))
C2182      ZZ=ZB1(JP2)-ZC(IC)
C2192      RHO2=XX*XX+YY*YY+ZZ*ZZ
C2202      RR=SQRT(RHO2)
C2212      A3=AIC*AJB*CEXP(-U*BK*RR)/4./PI/RR
C2222      CAP-EDGE(BOT) IMPEDANCE ELEMENTS COMPUTED.
C2232      Z(L3) = ZTZ , Z(L4) = ZRZ
C2242      Z3E=Z3E+TP(IP7)*TCT(IR7)/RHOC1(IR2)*TZ(JP7)*TPBE(JZ7)*A3
C2252      Z4E=Z4E+TZ(IP7)*(TPCR(IR7)/RC1(IP2)+TCR(IR7)/RI)*
C2262      1 TZ(JP7)*TPBE(JZ7)*A3
C2272      171 CONTINUE
C2282      170 CONTINUE
C2292      Z(L3)=U*ETA*(BK*Z3-Z3E/BK)
C2302      Z(L4)=U*ETA*(BK*Z4-Z4E/BK)
C2312      131 CONTINUE
C2322      130 CONTINUE
C2332      C EDGE-EDGE LOOP
C2342      DD 230 JC=1,NC
C2352      DD 230 JP=1,NM
C2362      L4=((LC2+LE)*(JC-1)+NM+(JP-1))*(LC2+LE)+LC2
C2372      DD 231 IC=1,NC
C2382      DD 231 IP=1,NM
C2392      L4=L4+1
C2402      Z4E=0.0
C2412      JP=2+(JP-1)
C2422      J7=4+(JP-1)
C2432      DD 270 JJP=1,4
C2442      JP2=JP2+1
C2452      JP7=JP7+1
C2462      JR2=2+LR
C2472      JR7=2+(JC-1)
C2482      JZ2=2+(JC-1)
C2492      JZ7=2+(JC-1)
C2502      DD 270 JJR2=1,2
C2512      JR2=JR2+1
C2522      JR7=JR7+1
C2532      JZ2=JZ2+1
C2542      JZ7=JZ7+1
C2552      IP2=2+(IP-1)
C2562      IP7=4+(IP-1)
C2572      DD 271 IIP=1,4
C2582      IP2=IP2+1
C2592      IP7=IP7+1
C2602      IR2=2+LR
C2612      IR7=2+(IC-1)

```

```

02584      IZ2=2*(IC-1)
02586      IZ7=2*(IC-1)
02592      DD 271 IIRZ=1,2
02602      IR2=IR2+1
02604      IR7=IR7+1
02606      IZ2=IZ2+1
02608      IZ7=IZ7+1
02610      AJC=AC(JP2)*ABS(RHOC(JR2+1)**2-RHOC(JR2)**2)
02612      AIC=AC(IP2)*ABS(RHOC(IR2+1)**2-RHOC(IR2)**2)
02614      AJB=DH(JP2)*ABS(ZE(JC)-ZC(JC))/2.
02616      AIB=DH(IP2)*ABS(ZE(IC)-ZC(IC))/2.
02618      IF(IC.NE.JC) GO TO 260
02620      IF(IR2.NE.JR2) GO TO 260
02622      IF(YB1(IP2).NE.YB1(JP2)) GO TO 260
02624      IF(XB1(IP2).NE.XB1(JP2)) GO TO 260
02626      C SELF TERM.
02628      A3=GS(IR2+(IP2-1)*NPR1)
02630      GO TO 265
02632      C NON-SELF TERM.
02634      260 CONTINUE
02636      XX=RHOC1(JR2)*RC1(JP2)*CPC(JP2)-RHOC1(IR2)*RC1(IP2)*CPC(IP2)
02638      YY=RHOC1(JR2)*RC1(JP2)*SPC(JP2)-RHOC1(IR2)*RC1(IP2)*SPC(IP2)
02640      ZZ=ZC(JC)-ZC(IC)
02642      RHQ2=XX*XX+YY*YY+ZZ*ZZ
02644      RR=SQRT(RHQ2)
02646      A3=AIC*AJC*CEXP(-U*BK*RR)/4./PI/RR
02648      265 CONTINUE
02650      RI=RHOC1(IR2)*RC1(IP2)
02652      RJ=RHOC1(JR2)*RC1(JP2)
02654      C EDGE (CAP)-EDGE (CAP) IMPEDANCE ELEMENTS COMPUTED.
02656      Z4E=Z4E+TZ(IP7)*T7(JP7)*((CPC(IP2)*CPC(JP2)+
02658      1 SPC(IP2)*SPC(JP2))*TCE(IR7)*TCE(JR7)-
02660      2 (TPCE(IR7)/RC1(IP2)+TCE(IR7)/RI)*
02662      3 (TPCE(JR7)/RC1(JP2)+TCE(JR7)/RJ)/BK/BK+A3
02664      XX=XB1(IP2)-(XC+RHOC1(IR2)*RC1(IP2)*CPC(IP2))
02666      YY=YB1(JP2)-(YC+RHOC1(JR2)*RC1(JP2)*SPC(JP2))
02668      ZZ=ZB(JZ2)-ZC(JC)
02670      RHQ2=XX*XX+YY*YY+ZZ*ZZ
02672      RR=SQRT(RHQ2)
02674      A3=AIC*AJB*CEXP(-U*BK*RR)/4./PI/RR
02676      C EDGE (CAP)-EDGE (BOT) IMPEDANCE ELEMENTS COMPUTED.
02678      Z4E=Z4E-TZ(IP7)*TPBE(JZ7)/BK/BK+A3
02680      1 TZ(JP7)*TPBE(JZ7)/BK/BK+A3
02682      XX=XB1(IP2)-(XC+RHOC1(JR2)*RC1(JP2)*CPC(JP2))
02684      YY=YB1(IP2)-(YC+RHOC1(JP2)*RC1(JP2)*SPC(JP2))
02686      ZZ=ZB(JZ2)-ZC(JC)
02688      RHQ2=XX*XX+YY*YY+ZZ*ZZ
02690      RR=SQRT(RHQ2)
02692      A3=AIB*AJC*CEXP(-U*BK*RR)/4./PI/RR
02694      C EDGE (BOT)-EDGE (CAP) IMPEDANCE ELEMENTS COMPUTED.
02696      Z4E=Z4E-TZ(IP7)*TPBE(IJ7)*TZ(JP7)*(TPCE(JR7)/RC1(JP2)+
02698      1 TCE(JR7)/RJ)/BK/BK+A3

```

```

02860      XX=XB1(JP2)-XB1(IP2)
02862      YY=YB1(JP2)-YB1(IP2)
02864      ZZ=ZB1(JP2)-ZB1(IP2)
02866      RHO2=XX*XX+YY*YY+ZZ*ZZ
02868C CHECK FOR SELF-TERM.
02870      IF(RHO2.EQ.C,0) RHO2=DH(IP2)*DH(IP2)/4.
02872      RR=SQRT(RHO2)
02874      A3=ATB*AJB*CEXP(-U*BK*RR)/4./PI/RR
02875C EDGE(BOT)-EDGE(BOT) IMPEDANCE ELEMENTS COMPUTED.
02876      Z4E=Z4E+TZ(IP7)*TZ(JP7)*TDE(IJ7)*Y5E(IJ7)-
02878      1 TPBE(IJ7)*TPBE(IJ7)/BK/BK)*A3
02992 271 CONTINUE
03002 270 CONTINUE
03042      Z(L4)=U*BK*ETA*Z4E
03052 231 CONTINUE
03062 230 CONTINUE
03063C USE SYMMETRY TO FILL EDGE-CAP MATRIX, FROM CAP-EDGE MATRIX.
03064      K1=LC2+1
03065      K2=LC2+LE
03066      DO 300 J=K1,K2
03067      DO 300 I=1,LC2
03068      L=(J-1)*K2+I
03069      LT=(I-1)*K2+J
03070      Z(LT)=Z(L)
03071 300 CONTINUE
03072C **** END CALCULATION OF ZCC MATRIX ****
03080 900 CONTINUE
03090C PRINT ZCC MATRIX BY SUBMATRIX BY COLUMNS.
03100      LI(1)=LC
03110      LI(2)=LC
03120      LI(3)=LE
03130      LJ(1)=LC
03140      LJ(2)=LC
03150      LJ(3)=LE
03160      CALL ZLIST(3,3,LI,LJ,NAMI,NAMJ,Z)
03170C WRITE ZCC TO DISK
03172      WRITE(7) (Z(I),I=1,LCC)
03174      STOP
03176      END
03340C -----
03345      SUBROUTINE BOTIN
03350C
03355C READ BOT COORDINATES AND COMPUTE BOT SEGMENT ARRAYS.
03360C
03365      COMMON /BOT1/ NMODE,NPT,NBAND
03370      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
03375      COMMON /BOT3/ DH(82),SV(82),CV(82)
03380      COMMON /BOT5/ T(160),TP(160),TZ(160)
03385      COMMON /BOT6/ TEDGE,IUNIF
03390      READ(5,49) NMODE,NPT,NBAND
03395      READ(5,49) NP
03400 49 FORMAT(3I3)

```

```

03405 WRITE(6,48) NMODE,NPT,NBAND,NP
03410 48 FORMAT(32H NMODE NPT NBAND NP,/,4I8)
03415 READ(5,53){YB(I),I=1,NP)
03420 53 READ(5,53){XB(I),I=1,NP)
03425 53 FORMAT(10F8.4)
03430 WRITE(6,55)
03435 55 FORMAT(7,3H YB)
03440 WRITE(6,46){YB(I),I=1,NP)
03445 46 FORMAT(1X,10F8.4)
03450 WRITE(6,56)
03455 56 FORMAT(7,3H XB)
03460 WRITE(6,46){XB(I),I=1,NP)
03465 C PLOT THE BODY COORDINATES.
03470 CALL PLOTB(XB,YB,NP,41)
03475 READ(5,53) BL
03480 WRITE(6,47) BL
03485 47 FORMAT(7,21H HALF-LENGTH OF BOT =,F12.4)
03490 C CHECK IF BOT CONTAINS AN ODD NUMBER OF POINTS.
03495 IF(MOD(NP,2).NE.1) GO TO 980
03500 C CHECK IF GENERATING CURVE IS OPEN OR CLOSED.
03505 IEDEGE=1
03510 IF((YB(1)-YB(NP)).NE.0..OR.(XB(1)-XB(NP)).NE.0.) GO TO 10
03515 IEDEGE=0
03520 YB(NP+1)=YB(2)
03525 XB(NP+1)=XB(2)
03530 YB(NP+2)=YB(3)
03535 XB(NP+2)=XB(3)
03540 NP=NP+2
03545 WRITE(6,66) NP
03550 66 FORMAT(7,39H BOT GENERATING CURVE IS CLOSED. NP = ,I3)
03555 10 CONTINUE
03560 C COMPUTATION OF BODY SEGMENT PARAMETERS.
03565 DO 57 I=2,NP
03570 I2=I-1
03575 RR1=YB(I1)-YB(I2)
03580 RR2=XB(I1)-XB(I2)
03585 DH(I2)=SQRT(RR1*RR1+RR2*RR2)
03590 XB1(I2)=.5*(XB(I1)+XB(I2))
03595 YB1(I2)=.5*(YB(I1)+YB(I2))
03600 SV(I2)=RR1/DH(I2)
03605 CV(I2)=RR2/DH(I2)
03610 57 CONTINUE
03615 C CHECK IF BOT SEGMENTATION IS UNIFORM.
03620 IUNIF=0
03625 NP1=NP-1
03630 DO 60 I=2,NP1
03635 RR1=DH(I)/DH(1)
03640 IF(RR1.L1.0.99 .OR. RR1.GT.1.01) GO TO 20
03645 60 CONTINUE
03650 IUNIF=1
03655 WRITE(6,67)
03660 67 FORMAT(7,1 BOT GENERATING CURVE HAS UNIFORM SEGMENTATION')

```

```

03665 20 CONTINUE
03670C COMPUTATION OF TRIANGLE FUNCTIONS T.
03675 NM=(NP-3)/2
03680 DO 74 J=1,NM
03685 J2=2*(J-1)+1
03690 J3=J2+1
03695 J4=J3+1
03700 J5=J4+1
03705 J6=4*(J-1)+1
03710 J7=J6+1
03715 J8=J7+1
03720 J9=J8+1
03725 DEL1=DH(J2)+DH(J3)
03730 DEL2=DH(J4)+DH(J5)
03735 TP(J6)=1./DEL1
03740 TP(J7)=1./DEL1
03745 TP(J8)=-1./DEL2
03750 TP(J9)=-1./DEL2
03755 T(J6)=DH(J2)/(2./DEL1)
03760 T(J7)=DH(J3)/(2./DEL1)
03765 T(J8)=(DH(J4)/2.+DH(J5))/DEL2
03770 T(J9)=(DH(J4)/2.-DH(J5))/DEL2
03775 74 CONTINUE
03780 NM4=NM+4
03785 DO 75 J=1,NM4
03790 T2(J)=T(J)
03795 75 IF (JEDGE.EQ.0) GO TO 76
03800 T2(1)=2.-T(1)
03805 T2(2)=2.-T(2)
03810 T2(NM4-1)=2.-T(NM4-1)
03815 T2(NM4)=2.-T(NM4)
03820 76 CONTINUE
03825 RETURN
03830 980 WRITE(6,981)
03835 981 FORMAT(/, ' **** ERROR IN NOT INPUT')
03840 STOP
03845 END
03850C *****
03855 SUBROUTINE PLNTR(X,Y,N,NP)
03860C *****
03865C
03870C WRITTEN 2/14/74 BY J. M. PUTNAM DEPT 226 X23877
03875C
03880C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
03885C N IS THE NUMBER OF POINTS TO BE PLOTTED.
03890C NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
03895C NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
03900C NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
03905C
03910 REAL X(1),Y(1),HEAD(10)
03915 INTEGER LINE(10),BLANK,STAR,PLUS
03920 DATA BLANK,STAR,PLUS /1H,1H*,1H+/

```

```

03925      NC=51
03930      N10=(NC-1)/10
03935      WRITE(6,500)
03940 500      FORMAT(//,44H BODY COORDINATES, + INDICATES TRIANGLE PEAK)
03945      WRITE(6,504)
03950      XMIN=X(1)
03955      XMAX=X(1)
03960      YMIN=Y(1)
03965      YMAX=Y(1)
03970      DO 6 I=1,N
03975      IF(X(I).LT.XMIN) XMIN=X(I)
03980      IF(X(I).GT.XMAX) XMAX=X(I)
03985      IF(Y(I).LT.YMIN) YMIN=Y(I)
03990      IF(Y(I).GT.YMAX) YMAX=Y(I)
03995 6      CONTINUE
04000      DEL=XMAX-XMIN
04005      IF(YMAX-YMIN.GT.DEL) DEL=YMAX-YMIN
04010      XMAX=XMIN+DEL
04015      YMAX=YMIN+DEL
04020      DO 5 I=1,N10
04025      Z=1
04030 5      HEAD(I)=(XMAX-XMIN)*Z/N10+XMIN
04035      DY=(YMAX-YMIN)/(NR-1)
04040      IEDGE=1
04045      IF(X(1).EQ.X(N) .AND. Y(1).EQ.Y(N)) IEDGE=0
04050      Z=YMAX+DY
04055      YL=Z-DY/2.
04060      DO 7 J=1,NR
04065      DO 8 K=1,NC
04070 8      LINE(K)=BLANK
04075      Z=Z-DY
04080      YU=YL
04085      YL=Z-DY/2.
04090      DO 9 I=1,N
04095      IF(Y(I).GE.YU) GO TO 9
04100      IF(Y(I).LT.YL) GO TO 9
04105      K=(X(I)-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
04110      IF(K.GT.NC) K=NC
04115      LINE(K)=STAR
04120      IF(MOD(I,2).EQ.1) LINE(K)=PLUS
04125      IF(IEGE.EQ.0) GO TO 9
04130      IF(I.EQ.1 .OR. I.EQ.N) LINE(K)=STAR
04135 9      CONTINUE
04140      WRITE(6,508) Z,(LINE(K),K=1,NC)
04145 7      CONTINUE
04150      WRITE(6,504)
04155      WRITE(6,3002)
04160      WRITE(6,507) XMIN,(HEAD(I),I=1,N10)
04165      RETURN
04170 304      FORMAT ( 1X, 14(1H-), 1H., 10(5H----.), 1H- )
04175 507      FORMAT(10X,11(F10.4))
04180 508      FORMAT (1X, F12.4,1X, 1HI, 51A1, 1HI )

```

```

04185 3002 FORMAT(4X,7HYH / XM,4X,1HI,5(9X,1HI))
04190C END
04195C -----
04200C SUBROUTINE CAPIN
04205C
04210C READ CAP INPUTS AND COMPUTE CAP ARRAYS.
04215C
04220C COMMON /BOT2/ NP,BL,YR(83),XB(83),YB1(82),XB1(82)
04225C COMMON /CAP1/ NC,XC,YC,ZC(2)
04230C COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
04235C COMMON /CAP3/ TCP(36),TCT(36),TPCR(36),TPCT(36)
04240C COMMON /CAP4/ RC(83),PC1(82),AC(82),CPC(82),SPC(82)
04245C COMMON /EDG1/ NE,ZE(2),ZRE(10)
04250C COMMON /EDG2/ TCE(10),TPCE(10),TBE(10),TPBE(10)
04255C READ(5,1) NC,NPR,NE
04260C 1 FORMAT(3I3)
04265C IF(NE.NE.0) NE=NC
04270C WRITE(6,3) NC,NPR,NE
04275C 3 FORMAT(16H1 NC NPR NE,/,3I5,/)
04280C IF(NC.EQ.0) RETURN
04285C READ(5,2) XC,YC
04290C 2 FORMAT(1CF8.4)
04295C READ(5,2) (ZC(I),I=1,NC)
04300C READ(5,2) (RHOC(I),I=1,NPR)
04305C IF(NE.NE.0) READ(5,2) (ZE(I),I=1,NE)
04310C WRITE(6,4)
04315C 4 FORMAT(37H CAP XC YC ZC ZE,/)
04320C DO 100 I=1,NC
04325C IF(NE.EQ.0) WRITE(6,5) I,XC,YC,ZC(I)
04330C IF(NE.NE.0) WRITE(6,5) I,XC,YC,ZC(I),ZE(I)
04335C 100 CONTINUE
04340C 5 FORMAT(14,4X,4F8.4)
04345C WRITE(6,6)
04350C 6 FORMAT(7,5H RHOC)
04355C WRITE(6,7) (RHOC(I),I=1,NPR)
04360C 7 FORMAT(1X,10F8.4)
04365C IF(MOD(NPR,2).NE.1) GO TO 980
04370C DO 120 I=2,NPR
04375C 120 IF(RHOC(I).LE.RHOC(I-1)) GO TO 980
04380C COMPUTATION OF CAP SECTOR PARAMETERS.
04385C DO 47 I=1,NP
04390C RC(I)=SQRT((YR(I)-YC)**2+(XB(I)-XC)**2)
04395C 47 CONTINUE
04400C DO 37 I=2,NP
04405C I2=I-1
04410C RR1=YB1(I2)-YC
04415C RR2=XB1(I2)-XC
04420C RC1(I2)=SQRT(RR1*RR1+RR2*RR2)
04425C AC(I2)=ABS((XB(I)-XC)*(YR(I)+YC)+(XB(I2)-XB(I))*
04430C 1 (YB(I2)+YB(I))+(XC-XB(I2))*(YC+Y4(I2)))/2.
04435C SPC(I2)=RR1/RC1(I2)
04440C CPC(I2)=RR2/RC1(I2)

```

```

04445 57 CONTINUE
04450 DO 67 I=2,NPR
04455 I2=I-1
04460 RHOC(I,I2)=(RHOC(I)+PHOC(I2))/2.
04465 DRHOC(I2)=RHOC(I)-PHOC(I2)
04470 67 CONTINUE
04475C COMPUTATION OF CAP TRIANGLE FUNCTIONS.
04480 LC=(NPR-3)/2
04485 DO 74 J=1,LC
04490 J2=2*(J-1)+1
04495 J3=J2+1
04500 J4=J3+1
04505 J5=J4+1
04510 J6=4*(J-1)+1
04515 J7=J6+1
04520 J8=J7+1
04525 J9=J8+1
04530 DEL1=DRHOC(J2)+DRHOC(J3)
04535 DEL2=DRHOC(J4)+DRHOC(J5)
04540 TPCR(J6)=1./DEL1
04545 TPCR(J7)=1./DEL1
04550 TPCR(J8)=-1./DEL2
04555 TPCR(J9)=-1./DEL2
04560 TCR(J6)=DRHOC(J2)/2./DEL1
04565 TCR(J7)=(DRHOC(J2)+DRHOC(J3))/2./DEL1
04570 TCR(J8)=(DRHOC(J4)/2.+DRHOC(J5))/DEL2
04575 TCR(J9)=DRHOC(J5)/2./DEL2
04580 74 CONTINUE
04585 LC4=LC+4
04590 DO 75 I=1,LC4
04595 TCT(I)=TCR(I)
04600 TPCT(I)=TPCR(I)
04605 75 CONTINUE
04610 TCT(LC4-1)=2.-TCT(LC4-1)
04615 TCT(LC4)=2.-TCT(LC4)
04620 TPCT(LC4-1)=-TPCT(LC4-1)
04625 TPCT(LC4)=-TPCT(LC4)
04630 IF(RHOC(1).EQ.0.0) GO TO 76
04635 TCT(1)=2.-TCT(1)
04640 TCT(2)=2.-TCT(2)
04645 TPCT(1)=-TPCT(1)
04650 TPCT(2)=-TPCT(2)
04655 76 CONTINUE
04660 IF(NC.EQ.0) RETURN
04665C COMPUTATION OF EDGE HALF TRIANGLE FUNCTIONS.
04670 DO 76 IC=1,NC
04675 J2=2*IC-2
04680 J3=J2+1
04685 J6=2*(IC-1)+1
04690 J7=J6+1
04695 DEL1=DRHOC(J2)+DRHOC(J3)
04700 TPCE(J6)=1./DEL1

```

```

04765      TPCE(J7)=1./DEL1
04770      TCCE(J6)=DRHOC(J2)/2./DEL1
04775      TCCE(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
04780      DEL2=ZE(IC)-ZC(IC)
04785      ZBCE(J6)=ZC(IC)+0.25*DEL2
04790      ZBCE(J7)=ZC(IC)+0.75*DEL2
04795      IF(DEL2.LT.0.0) GO TO 78
04800 C EDGE IS AT Z=-L.
04805      TPBE(J6)=-1./DEL2
04810      TPBE(J7)=-1./DEL2
04815      TBCE(J6)=0.75
04820      TBCE(J7)=0.25
04825      GO TO 8C
04830      78 CONTINUE
04835 C EDGE IS AT Z=+L.
04840      TPBE(J6)=1./DEL2
04845      TPBE(J7)=1./DEL2
04850      TBCE(J6)=-0.75
04855      TBCE(J7)=-0.25
04860      8C CONTINUE
04865      RETURN
04870      98C WRITE(6,981)
04875      981 FORMAT(/,' **** ERROR IN CAP INPUT')
04880      STOP
04885      END
06212 C *****
06222 SUBROUTINE CSIMP(F,A,B,DEL,IMAX,S11,S,N,IER)
06232 C *****
06242 C COMPLEX SIMPSON INTEGRATION ROUTINE.
06252 COMPLEX F,S11,S,SUMK
06262 S11=(0.,0)
06272 S=(.C,.C)
06282 N=0
06292 BA=B-A
06302 IF (BA) 20,19,20
06312 19 IER=1
06322 RETURN
06332 20 IF(DEL) 22,22,23
06342 22 IER=2
06352 RETURN
06362 23 IF(IMAX-1) 24,24,25
06372 24 IER=3
06382 RETURN
06392 25 X=BA/2.,+A
06402 NHALF=1
06412 SUMK=F(X)+BA*2./3.
06422 S=SUMK+(F(A)+F(B))*BA/6.
06432 DO 28 I=2,IMAX
06442 S11=S
06452 S=(S-SUMK/2.)/2.
06462 NHALF=NHALF*2
06472 ANHALF=NHALF

```

```

06482 FRSTX=A*(BA/ANHLF)/2.
06492 SUMK=F(FRSTX)
06502 XK=FRSTX
06512 KLAST=NHOLF-1
06522 FINC=BA/ANHLF
06532 DO 26 K=1,KLAST
06542 XK=XK+FINC
06552 26 SUMK=SUMK+F(XK)
06562 SUMK=SUMK*2.*BA/(3.*ANHLF)
06572 S=S+SUMK
06582 IF(CABS(S).EQ.0.0) GO TO 29
06592 IF((CABS(S-SI1)/CABS(S))-DEL) 29,28,28
06602 28 CONTINUE
06612 IER=4
06622 WRITE(6,1) SI1,S
06632 1 FORMAT(/,29H INTEGRATION DID NOT CONVERGE,/,
06642 1 29H THE PREVIOUS AND FINAL VALUES OF THE INTEGRAL ARE,
06652 2 13H RESPECTIVELY,4E12.3)
06662 GO TO 30
06672 29 IER=C
06682 30 N=2*NHOLF
06692 RETURN
06702 END
-----
06712C COMPLEX FUNCTION FUNC(R)
06722 COMPLEX U
06732 COMMON /WAVE/ BK
06742 COMMON /INT/ CONST1,CONST2
06752 DATA U /10.,1./
06762 T=SQRT(CONST1+R*R+P*CONST2)
06772 FUNC=R*CEXP(-U*BK*T)/T
06782 RETURN
06792 END
-----
06810C SUBROUTINE ZLIST(NI,NJ,LI,LJ,NAMI,NAMJ,Z)
06820 WRITE(2,1) NI,NJ,LI,LJ,NAMI,NAMJ,Z
06830C 1 WRITE(2,1) NI,NJ,LI,LJ,NAMI,NAMJ,Z
06840 COMPLEX Z(1)
06850 DIMENSION LI(1),LJ(1),NAMI(1),NAMJ(1)
06860 NCOL=0
06870 DO 100 I=1,NI
06880 100 NCOL=NCOL+LI(I)
06890 NR=1
06900 DO 300 J=1,NJ
06910 IF(LJ(J).EQ.0) GO TO 300
06920 NC=1
06930 DO 200 I=1,NI
06940 IF(LI(I).EQ.0) GO TO 200
06950 WRITE(6,1) NAMI(I),NAMJ(J)
06960 1 FORMAT(1,2H Z,1X,A4,3H - ,A4)
06970 KI=LI(I)
06980 KJ=LJ(J)
06990 K1=(NR-1)*NCOL+NC

```

```

07000 DO 150 KK=1,KJ
07010 K2=K1+K1-1
07020 WRITE(6,2) (Z(K),K=K1,K2)
07030 2 FORMAT(1X,10G11.4)
07040 150 K1=K1+NC0
07050 NC=NC+1
07060 200 CONTINUE
07070 NR=NR+L(J)
07080 300 CONTINUE
07090 RETURN
07100 END
07110/END

```

1 1 1 1

```

CC100RDTZ,P3C,T100,CH140000,STCX3.
CC110ACCOUNT,M03265,BOT2A.
CC120BANNERS(OUTPUT)*J. PUTNAM*DEPT 220*BLD 110-4**
CC130DEFINE(OUTFIL=ZSWX)
CC138FTN(R=0,DPT=2)
CC140LDSET(PRESET=INDEF)
CC150LGO.
CC210EXIT.
CC230/ FOR
CC240      PROGRAM BOTZSW(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
CC250      1 OUTFIL,TAPE7=OUTFIL)
CC252C
CC254C BOTZSW GENERATES THE BOT-WIRE/JUNCTION MATRIX.
CC256C
CC259C UNIT 5 IS THE CARD READER.
CC260C UNIT 6 IS THE LINE PRINTER.
CC262C UNIT 7 IS A DISK FILE FOR OUTPUT OF THE Z MATRIX.
CC264C
CC280      COMPLEX A3,Z(4400),G1(3403),G2(3403),U
CC281      COMPLEX G1J,G2J
CC282      COMMON /WAVE/ BK
CC284      COMMON /BOT1/ NMODE,NPT,NBAND
CC286      COMMON /BOT2/ NP,RL,YR(83),XB(83),YB1(82),XB1(82)
CC288      COMMON /BOT3/ DH(82),SV(82),CV(82)
CC290      COMMON /BOT5/ T(160),TP(160),TZ(160)
CC291      COMMON /BOT6/ IEDGE,IUNIT
CC292      COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
CC293      1 XW1(100),YW1(100),ZW1(100)
CC294      COMMON /WIRE2/ DHW(100),DXW(100),DYW(100),DZW(100)
CC295      COMMON /WIRE3/ NW,INOW(6),RADW(5)
CC296      COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
CC297      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
CC298      COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(16)
CC302      COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
CC304      COMMON /JUNC4/ UXJ(10),UYJ(10),UZJ(10)
CC306      COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
CC308      COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
CC320      COMMON /INT/ M,RH02,ZP
CC325      DIMENSION LI(2),LJ(2),NAMI(2),NAMJ(2)
CC330      EXTERNAL FUNC1,FUNC2
CC340      DATA NAMI,NAMJ /4HS(T),4HS(Z),4HWIRE,4HJUNC/
CC430      U=10.,1.)
CC432      PI=3.14159265
CC434      ETA=376.707
CC435      READ(5,1) BK
CC436 1    FORMAT(E15.7)
CC437      WRITE(6,2) BK
CC438 2    FORMAT(9H1      BK,/,E15.7)
CC440C
CC450      CALL BOTIN
CC460C
CC470      KG=NP-1

```

```

00480      NM=(NP-3)/2
00490      NM2=NM*2
00500      BK2=BK*BK
00510      BKL=BK*BL
00790C
00800      CALL SCAPIN
00835C
00836      CALL WIREIN
00837C
00838      WRITE(7) NP,NW,NPW,NJ
00840      KGW=NPW-1
00842      LSW=NM2*LV
00844      LSWJ=NM2*(LV+NJ)
00860C  CONSTANTS USED IN SUMMATION OF IMPEDANCE ELEMENTS.
00865      C1=BK*ETA
00870      C2=C1
00890C  *** END INITIALIZATION OF PARAMETERS ***
00939C  COMPUTATION OF IMPEDANCE MATRICES Z(M), WHERE M = -NMODE+1 TO
00941C  NMODE-1.
00943      N=-NMODE
00947      KMODE=2*NMODE-1
00962      DO 998 MM=1,KMODE
00963      M=M+1
00965      WRITE(6,54) M
00970  54  FORMAT(4H) M=,I3)
01000C  COMPUTATION OF GREEN'S FUNCTION KERNEL.
01010      I1=0
01020      NPT1=0
01030      NINT1=0
01040      NPT2=0
01050      NINT2=0
01188      IW=1
01190      DO 16 J=1,KGW
01192C  DETERMINE IF SEGMENT J CONNECTS 2 DISTINCT WIRES. IF SO, SKIP SEGMENT J.
01200      IF(J+1.EQ.INDW(IW)) GO TO 7
01230      DO 17 I=1,KG
01240      I1=I1+1
01250      YY=YB1(I1)-YB1(J)
01260      XX=XB1(I1)-XB1(J)
01270      RMD2=XX*XX+YY*YY
01280      ZP=ZW1(J)
01400      CALL CSIMP(FUNC2,-PL,BL,0.05,10,A3,G2(I1),K,IER)
01402      NPT2=NPT2+K+1
01404      NINT2=NINT2+1
01405      G2(I1)=G2(I1)/4./PI
01410      IF(MP.NE.1) GO TO 17
01420      CALL CSIMP(FUNC1,-RL,BL,0.05,10,A3,G1(I1),K,IER)
01430      NPT1=NPT1+K+1
01440      NINT1=NINT1+1
01450      G1(I1)=G1(I1)/4./PI
01650      17  CONTINUE
01652      GO TO 16

```

```

01654 7 I1=I1+KG
01656 IW=IW+1
01660 16 CONTINUE
01661 AVG1=C.0
01662 IF(NP.EQ.1) AVG1=NPT1/FLDPT(NINT1)
01664 AVG2=NPT2/FLDPT(NINT2)
01666 WRITE(6,56) NINT1,AVG1,NINT2,AVG2
01668 56 FORMAT(/,' NUMBER OF INTEGRATIONS AND AVERAGE NUMBER OF ',
01670 1 ' POINTS PER INTEGRATION (G1 AND G2, RESPECTIVELY)',/,
01672 2 2(I8,G1C.3))
01690C *** END CALCULATION OF GREEN'S KERNEL ***
02005C COMPUTATION OF IMPEDANCE ELEMENTS.
02007C BOT-WIRE LOOP.
02010 DO 30 J=1,LW
02020 JL=(J-1)*NM2
02030 J3=(J-1)*4
02040 J1=INDT(J)-1
02050 DO 31 I=1,NM
02055C COMPUTE SUBMATRIX INDICES.
02060 L1=JL+I
02070 L2=L1+NM
02080 Z(L1)=0.
02090 Z(L2)=0.
02100 I1=2*(I-1)
02110 I3=(I-1)*4
02120 DO 70 JJ=1,4
02130 J2=J1+JJ
02140 J7=J3+JJ
02150 J4=(J2-1)*KG+I1
02160 DO 71 II=1,4
02170 I2=I1+II
02180 I7=I3+II
02190 J4=J4+1
02200C BOT-WIRE IMPEDANCE ELEMENTS COMPUTED.
02210 Z(L1) = Z(L1) + Z(L2)
02220 Z(L1) = Z(L1) + DH(I2)*((CV(I2)*DXW(J2)+SV(I2)*DYW(J2))*
02230 1 T(I7)*TW(J7)-DHW(J2)*TP(I7)*TPW(J7)/BK2)+G2(J4)
02240 Z(L2) = Z(L2) + DH(I2)*TZ(I7)*(OZW(J2)*TW(J7)+G2(J4)-
02250 1 (-1)**NM*G1(J4))+U*H*PI*DHW(J2)*TPW(J7)/BK/BKL*G2(J4))
02260 71 CONTINUE
02270 70 CONTINUE
02280 Z(L1)=U*C1*Z(L1)
02290 Z(L2)=U*C2*Z(L2)
02300 31 CONTINUE
02310 30 CONTINUE
02320 IF(NJ.EQ.0) GO TO 900
02330C BOT-JUNCTION LOOP.
02340 DO 130 J=1,NJ
02350 JL=(J-1)*NM2+LSW
02360 J3=(J-1)*2
02370 J1=INDT(J)-1
02380 DO 131 I=1,NM

```

```

Q2477C COMPUTE SUBMATRIX INDICES.
Q2487 L1=J1+I
Q2497 L2=L1+NM
Q2507 Z(L1)=0.
Q2517 Z(L2)=0.
Q2527 I1=2*(I-1)
Q2537 I3=(I-1)*4
Q2547 DO 17C JJ=1,2
Q2557 J2=J1+JJ
Q2567 J7=J3+JJ
Q2577 J4=(J2-1)*KG+I1
Q2587 DO 171 II=1,4
Q2597 I2=I1+II
Q2607 I7=I3+II
Q2617 J4=J4+1
Q2627C BOT-JUNCTION(WIRE) IMPEDANCE ELEMENTS COMPUTED.
Q2629C Z(L1) = ZT, Z(L2) = ZZ.
Q2637 Z(L1)=Z(L1)+DH(I2)*((CV(I2)*DXW(J2)+SV(I2)*DYW(J2))*
Q2647 1 T(I1)+TJ(J7)-DHW(J2)*T*(I7)+TPJ(J7)/BK2)*G2(J4)
Q2657 Z(L2)=Z(L2)+DH(I2)*TZ(I7)*(DZW(J2)+TJ(J7)*G2(J4)-
Q2667 1 (-1)*M*G1(J4))+U*M*PI+DHW(J2)*TPJ(J7)/BK/BKL*G2(J4))
Q2668 171 CONTINUE
Q2670 170 CONTINUE
Q2672 DO 27C JJ=1,4
Q2674 ALPHA=PI*((JJ-1)/2.+0.25)
Q2676 SA=SIN(ALPHA)
Q2678 CA=CLS(ALPHA)
Q2680 RA=(RADJ(J)+RADD(J))/2.
Q2682 XA=XJ(J)+RA*(CA*UXJ1(J)+SA*UXJ2(J))
Q2684 YA=YJ(J)+RA*(CA*UYJ1(J)+SA*UYJ2(J))
Q2686 ZA=ZJ(J)+RA*(CA*UZJ1(J)+SA*UZJ2(J))
Q2688 DC 271 II=1,4
Q2690 I2=I1+II
Q2692 I7=I3+II
Q2694 YY=YB1(I2)-YA
Q2696 XX=XB1(I2)-XA
Q2698 RHO2=XX*XX+YY*YY
Q2700 ZP=ZA
Q2702 CALL CSIMP(FUNC1,-BL,BL,0.05,10,A3,G1J,K,IER)
Q2704 G1J=G1J/4./PI
Q2706 CALL CSIMP(FUNC2,-BL,BL,0.05,10,A3,G2J,K,IER)
Q2708 G2J=G2J/4./PI
Q2709 DT=CA*(CV(I2)*UXJ1(J)+SV(I2)*UYJ1(J))+
Q2710 1 SA*(CV(I2)*UXJ2(J)+SV(I2)*UYJ2(J))
Q2711 DZ=CA*UZJ1(J)+SA*UZJ2(J)
Q2712 RR=(RADJ(J)-RADD(J))/2.
Q2713C BOT-JUNCTION(DISK) IMPEDANCE ELEMENTS COMPUTED.
Q2714 Z(L1) = ZT, Z(L2) = ZZ.
Q2715 Z(L1)=Z(L1)+DH(I2)/4.*(DT+T(I7)*RR-
Q2716 1 TP(I7)/BK2)*G2J
Q2717 Z(L2)=Z(L2)+DH(I2)*TZ(I7)/4.*(DZ*RR*(G2J-(-1)*M*G1J)+
Q2718 1 U*M*PI/BK/BKL*G2J)

```

```

Q2718 271 CONTINUE
Q2719 270 CONTINUE
Q2720 Z(L1)=U*C1*Z(L1)
Q2722 Z(L2)=U*C2*Z(L2)
Q2724 131 CONTINUE
Q2726 130 CONTINUE
Q2737C *** END CALCULATION OF Z(M) SUBMATRIX ***
Q2740 900 CONTINUE
Q2747C PRINT OUT IMPEDANCE MATRIX Z(M) BY SUBMATRIX BY COLUMNS.
Q2800 LI(1)=NM
Q2805 LI(2)=NM
Q2810 LJ(1)=LW
Q2815 LJ(2)=MJ
Q2820 CALL ZLIST(2,2,LI,LJ,NAMI,NAMJ,Z)
Q2845C WRITE Z(M) TO DISK.
Q2850 WRITE(7) M
Q2860 WRITE(7) (Z(I),I=1,LSWJ)
Q2870 998 CONTINUE
Q2927 STOP
Q2937 END
-----
Q2950C SUBROUTINE BOTIN
Q2955C
Q2960C READ BOT COORDINATES AND COMPUTE BOT SEGMENT ARRAYS.
Q2970C
Q2975 COMMON /BOT1/ NM0DE,NPT,NBAND
Q2980 COMMON /BOT2/ NP,BL,YB(83),YB(83),YB(82),XB(82)
Q2985 COMMON /BOT3/ OH(82),SV(82),CV(82)
Q2990 COMMON /BOT5/ T(160),TP(160),TZ(160)
Q2995 COMMON /BOT6/ IEDGE,IUNIF
Q3000 READ(5,49) NM0DE,NPT,NBAND
Q3005 READ(5,49) NP
Q3010 49 FORMAT(3I3)
Q3015 WRITE(6,48) NM0DE,NPT,NBAND,NP
Q3020 48 FORMAT(32H NM0DE NPT NBAND NP,/,4I8)
Q3025 READ(5,53) (YB(I),I=1,NP)
Q3030 READ(5,53) (XB(I),I=1,NP)
Q3035 53 FORMAT(10F8.4)
Q3040 WRITE(6,55)
Q3045 55 FORMAT(1,3H YB)
Q3050 WRITE(6,46) (YB(I),I=1,NP)
Q3055 46 FORMAT(1X,10F8.4)
Q3060 WRITE(6,56)
Q3065 56 FORMAT(1,3H XB)
Q3070 WRITE(6,46) (XB(I),I=1,NP)
Q3075C PLOT THE BODY COORDINATES.
Q3080 CALL PLOT8(XB,YB,NP,41)
Q3085 READ(5,53) BL
Q3090 WRITE(6,47) BL
Q3095 47 FORMAT(1,21H HALF-LENGTH OF BOT =,F12.4)
Q3100C CHECK IF BOT CONTAINS AN ODD NUMBER OF POINTS.
Q3105 IF (MOD(NP,2).NE.1) GO TO 980

```

```

03110C CHECK IF GENERATING CURVE IS OPEN OR CLOSED.
031115 IEDGE=1
031120 IF((YB(1)-YB(NP)).NE.0..OR.(XB(1)-XB(NP)).NE.0.) GO TO 10
031125 IEDGE=0
031130 YB(NP+1)=YB(2)
031135 XB(NP+1)=XB(2)
031140 YB(NP+2)=YB(3)
031145 XB(NP+2)=XB(3)
031150 NP=NP+2
031155 WRITE(6,66) NP
031160 66 FORMAT(//,39H BOT GENERATING CURVE IS CLOSED. NP = ,I3)
031165 10 CONTINUE
031170C COMPUTATION OF BODY SEGMENT PARAMETERS.
031175 DO 57 I=2,NP
031180 I2=I-1
031185 RR1=YB(I)-YB(I2)
031190 RR2=XB(I)-XB(I2)
031195 DH(I2)=SQRT(RR1*RR1+RR2*RR2)
031200 XB1(I2)=.5*(XB(I)+XB(I2))
031205 YB1(I2)=.5*(YB(I)+YB(I2))
031210 SV(I2)=RR1/DH(I2)
031215 CV(I2)=RR2/DH(I2)
031220 57 CONTINUE
031225C CHECK IF BOT SEGMENTATION IS UNIFORM.
031230 IUNIF=0
031235 NP1=NP-1
031240 DO 60 I=2,NP1
031245 RR1=DH(I)/DH(1)
031250 IF(RR1.(V.0.99..OR. RR1.GT.1.01) GO TO 20
031255 60 CONTINUE
031260 IUNIF=1
031265 WRITE(6,67)
031270 67 FORMAT(//,1 BOT GENERATING CURVE HAS UNIFORM SEGMENTATION')
031275 20 CONTINUE
031280C COMPUTATION OF TRIANGLE FUNCTIONS T.
031285 NM=(NP-3)/2
031290 DO 74 J=1,NM
031295 J2=2*(J-1)+1
031300 J3=J2+1
031305 J4=J3+1
031310 J5=J4+1
031315 J6=4*(J-1)+1
031320 J7=J6+1
031325 J8=J7+1
031330 J9=J8+1
031335 DEL1=DH(J2)+DH(J3)
031340 DEL2=DH(J4)+DH(J5)
031345 TP(J6)=1./DEL1
031350 TP(J7)=1./DEL1
031355 TP(J8)=-1./DEL2
031360 TP(J9)=-1./DEL2
031365 T(J6)=DH(J2)/2./DEL1

```

```

03376      T(J7)=(DH(J2)+DH(J3)/2.)/DEL1
03377      T(J8)=(DH(J4)/2.+DH(J5))/DEL2
03380      T(J9)=DH(J5)/2./DEL2
03385      74 CONTINUE
03390      NM4=NM+4
03395      DO 75 J=1,NM4
03400      75 T(J)=T(J)
03405      IF(IEDGE.EQ.0) GO TO 76
03410      TZ(1)=2.-T(1)
03415      TZ(2)=2.-T(2)
03420      TZ(NM4-1)=2.-T(NM4-1)
03425      TZ(NM4)=2.-T(NM4)
03430      76 CONTINUE
03435      RETURN
03440      980 WRITE(6,981)
03445      981 FORMAT(/,' ***** ERROR IN BOT INPUT')
03450      STOP
03455      END
-----
03855C      SUBROUTINE SCAPIN
03856C
03857C      SKIP CAP INPUTS.
03858C
03859C      READ(5,1) NC,NPR,NE
03861      1 FORMAT(3I3)
03862      IF(NC.EQ.0) RETURN
03863      IF(NE.NE.0) NE=NC
03865      READ(5,2) XC,YC
03866      2 FORMAT(2F8.4)
03868      READ(5,2) (ZC,I=1,NC)
03869      READ(5,2) (PHC,I=1,NPR)
03870      IF(NE.EQ.0) RETURN
03871      READ(5,2) (ZE,I=1,NE)
03872      RETURN
03873      END
-----
03875C      SUBROUTINE WIREIN
03877C
03887C      READ WIRE COORDINATES AND COMPUTE WIRE SEGMENT ARRAYS.
03897C
03907C      COMMON /WIRE1/ NPH,XW(101),YW(101),ZW(101),
03917      1 XW1(100),YW1(100),ZW1(100)
03927      COMMON /WIRE2/ DHW(100),DXW(100),DYW(100),DZW(100)
03937      COMMON /WIRE3/ NW,INW(6),RADW(5)
03947      COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
03957      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
03967      COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
03977      COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
03987      COMMON /JUNC4/ UXJ(10),UYJ(10),UZJ(10)
03990      COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
03992      COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
03993      NW=0

```

```

03994      NJ=0
03995      LU=0
03997      READ(5,51) NW,NPW,NJ
04007 51    FORMAT(3I3)
04008      WRITE(6,1) NW,NPW,NJ
04009 1      FORMAT(24H1      NW      NPW      NJ,/,3I8)
04010      IF(NW.EQ.0) RETURN
04017      READ(5,53) (XW(I),I=1,NPW)
04027      READ(5,53) (YW(I),I=1,NPW)
04037      READ(5,53) (ZW(I),I=1,NPW)
04047 53    FORMAT(10F8.4)
04057      READ(5,52) (INDW(I),I=1,NW)
04060 52    FORMAT(10I8)
04067      INDW(NW+1)=NPW+1
04070      READ(5,53) (RADW(I),I=1,NW)
04072      IF(INJ.EQ.0) GO TO 50
04077C NOTE, INDJW MUST BE MONOTONIC INCREASING.
04078      READ(5,52) (INDJW(I),I=1,NJ)
04080      READ(5,53) (RADD(I),I=1,NJ)
04082      READ(5,53) (UXJ(I),I=1,NJ)
04084      READ(5,53) (UYJ(I),I=1,NJ)
04086      READ(5,53) (UZJ(I),I=1,NJ)
04097 50    IERW=0
04107      IERJ=0
04117      WRITE(6,61)
04120 61    FORMAT(1,22X,'WIRE COORDINATES',2CX,'JUNCTION PARAMETERS',/,
04122      1 20X,'IW',4X,'XW',6X,'YW',6X,'ZW',
04124      2 10X,'IJ',4X,'RADD',5X,'UXJ',5X,'UYJ',5X,'UZJ')
04137      IJ=1
04138C THIS LOOP LISTS WIRE/JUNCTION POINTS, WHILE CHECKING THE FOLLOWING:
04139C 1) EACH WIRE MUST CONTAIN AN ODD NUMBER OF POINTS.
04140C 2) EACH JUNCTION MUST EITHER START OR TERMINATE A WIRE.
04143C 3) CHECK THAT ALL JUNCTION POINTS ARE FOUND.
04147      DO 100 IW=1,NW
04157      WRITE(6,62) IW,RADW(IW)
04167 62    FORMAT(2X,'WIRE',I3,' RADW=',F8.4)
04177      I1=INDW(IW)
04187      I2=INDW(IW+1)-1
04197C CHECK FOR AN ODD NUMBER OF POINTS ON WIRE IW.
04207      IF(MOD(I2-I1+1,2).NE.1) IERW=1
04217      DO 90 I=I1,I2
04227      WRITE(6,63) I,XW(I),YW(I),ZW(I)
04237 63    FORMAT(25X,I3,3F8.4)
04247C CHECK IF WIRE POINT I IS A JUNCTION POINT.
04257      IF(IJ.GT.NJ) GO TO 90
04267      IF(INDJW(IJ).NE.I) GO TO 90
04270C CHECK THAT JUNCTION POINT IJ IS AT THE START OR END OF WIRE.
04272      IF(I.NE.I1 .AND. I.NE.I2) IERJ=1
04280      WRITE(6,64) IJ,RADD(IJ),UXJ(IJ),UYJ(IJ),UZJ(IJ)
04290 64    FORMAT(14,25X,I3,4F8.4)
04300      IERJ=1
04307C DETERMINE DIRECTION IN WHICH WIRE LEAVES JUNCTION POINT.
04317      INDJ(IJ)=1

```

```

C4397      IF(I.EQ.I2) INDTJ(IJ)=-1
C4398C COMPUTE JUNCTION PARAMETERS.
C4400      RADJ(IJ)=RADW(IW)
C4405      XJ(IJ)=XW(I)
C4406      YJ(IJ)=YW(I)
C4407      ZJ(IJ)=ZW(I)
C4408      IJ=IJ+1
C4477 90 CONTINUE
C4487 100 CONTINUE
C4497C CHECK FOR WIRE OR JUNCTION INPUT ERRORS.
C4507      IF(IERW.NE.0) GO TO 980
C4517      IF(IJ-1.NE.NJ .OR. IERJ.NE.0) GO TO 990
C4527C COMPUTATION OF WIRE SEGMENT PARAMETERS.
C4537      DO 57 I=2,NPW
C4547      I2=I-1
C4557      DXW(I2)=XW(I)-XW(I2)
C4567      DYW(I2)=YW(I)-YW(I2)
C4577      DZW(I2)=ZW(I)-ZW(I2)
C4587      DHW(I2)=SQRT(DXW(I2)**2+DYW(I2)**2+DZW(I2)**2)
C4588      XW(I2)=C.5*(XW(I)+XW(I2))
C4590      YW(I2)=C.5*(YW(I)+YW(I2))
C4592      ZW(I2)=C.5*(ZW(I)+ZW(I2))
C4597 57 CONTINUE
C4607C COMPUTATION OF WIRE TRIANGLE FUNCTIONS TW.
C4617      LW=0
C4627      DO 75 IW=1,NW
C4637      I1=INDW(IW)
C4647      I2=INDW(IW+1)-1
C4657      LW1=(I2-I1-2)/2
C4667      DO 74 J=1,LW1
C4677      LW=LW+1
C4687      J2=2*(J-1)+I1
C4697      J3=J2+1
C4707      J4=J3+1
C4717      J5=J4+1
C4727      J6=4*(LW-1)+1
C4737      J7=J6+1
C4747      J8=J7+1
C4757      J9=J8+1
C4767      INDTW(LW)=J2
C4777      DEL1=DHW(J2)+DHW(J3)
C4787      DEL2=DHW(J4)+DHW(J5)
C4797      TPW(J6)=1./DEL1
C4807      TPW(J7)=1./DEL1
C4817      TPW(J8)=-1./DEL2
C4827      TPW(J9)=-1./DEL2
C4837      TW(J6)=DHW(J2)/2./DEL1
C4847      TW(J7)=(DHW(J2)+DHW(J3))/2./DEL1
C4857      TW(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
C4867      TW(J9)=DHW(J5)/2./DEL2
C4877 74 CONTINUE
C4887 75 CONTINUE

```

```

C4897C NOTE, LM=(MPW-NW)/2-NW
C4900C IF(IJ.EQ.0) RETURN
C4907C COMPUTATION OF JUNCTION HALF TRIANGLE FUNCTIONS TJ.
C4917 DO 85 IJ=1,NJ
C4927 IF(INDJ(IJ).GT.0) GO TO 80
C4930C JUNCTION IS AT THE END OF A WIRE.
C4937 J2=INDJW(IJ)-2
C4947 J3=J2+1
C4957 J6=2*(IJ-1)+1
C4967 J7=J6+1
C4977 INDJ(IJ)=J2
C4987 DEL1=-DHW(J2)-DHW(J3)
C4997 TPJ(J6)=1./DEL1
C5007 TPJ(J7)=1./DEL1
C5017 TJ(J6)=DHW(J2)/2./DEL1
C5027 TJ(J7)=(DHW(J2)+DHW(J3))/2./DEL1
C5037 GO TO 85
C5040C JUNCTION IS AT THE START OF A WIRE.
C5047 80 J4=INDJW(IJ)
C5057 J5=J4+1
C5067 J8=2*(IJ-1)+1
C5077 J9=J8+1
C5087 INDJ(IJ)=J4
C5097 DEL2=DHW(J4)+DHW(J5)
C5107 TPJ(J8)=-1./DEL2
C5117 TPJ(J9)=-1./DEL2
C5127 TJ(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
C5137 TJ(J9)=DHW(J5)/2./DEL2
C5147 85 CONTINUE
C5157C COMPUTE UNIT VECTORS FOR JUNCTION(DISK).
C5167 DO 95 IJ=1,NJ
C5177C NORMAL UNIT VECTOR (UJ).
C5187 RR=SQRT(UXJ(IJ)**2+UYJ(IJ)**2+UZJ(IJ)**2)
C5197 UXJ(IJ)=UXJ(IJ)/RR
C5207 UYJ(IJ)=UYJ(IJ)/RR
C5217 UZJ(IJ)=UZJ(IJ)/RR
C5227C FIND 2 ORTHOGONAL UNIT VECTORS IN THE PLANE OF THE DISK (UJ1 & UJ2).
C5237 UXJ1(IJ)=0.0
C5247 UYJ1(IJ)=0.0
C5257 UZJ1(IJ)=0.0
C5267 UXJ2(IJ)=0.0
C5277 UYJ2(IJ)=0.0
C5287 UZJ2(IJ)=0.0
C5297 IF(UXJ(IJ).EQ.0.0) GO TO 91
C5307C FIND INTERSECTION WITH X-Y PLANE.
C5317 UYJ1(IJ)=1.0
C5327 UXJ1(IJ)=-UYJ(IJ)/UXJ(IJ)
C5337C FIND INTERSECTION WITH X-Z PLANE.
C5347 UZJ2(IJ)=1.0
C5357 UXJ2(IJ)=-UZJ(IJ)/UXJ(IJ)
C5367 GO TO 94
C5377 91 IF(UYJ(IJ).EQ.0.0) GO TO 92

```

```

C5387C FIND INTERSECTION WITH Y-Z PLANE.
C5397    UZJ1(IJ)=1.0
C5407    UYJ1(IJ)=-UZJ1(IJ)/UYJ1(IJ)
C5417C FIND INTERSECTION WITH X-Y PLANE.
C5427    UXJ2(IJ)=1.0
C5437    UYJ2(IJ)=-UXJ1(IJ)/UYJ1(IJ)
C5447    GO TO 94
C5457 92 IF(UZJ1(IJ),EQ,0.0) GO TO 94
C5467C FIND INTERSECTION WITH X-Z PLANE.
C5477    UXJ1(IJ)=1.0
C5487    UZJ1(IJ)=-UXJ1(IJ)/UZJ1(IJ)
C5497C FIND INTERSECTION WITH Y-Z PLANE.
C5507    UYJ2(IJ)=1.0
C5517    UZJ2(IJ)=-UYJ1(IJ)/UZJ1(IJ)
C5527 94 CONTINUE
C5537    RR=SQRT(UXJ1(IJ)**2+UYJ1(IJ)**2+UZJ1(IJ)**2)
C5547    UXJ1(IJ)=UXJ1(IJ)/RR
C5557    UYJ1(IJ)=UYJ1(IJ)/RR
C5567    UZJ1(IJ)=UZJ1(IJ)/RR
C5577    RR=SQRT(UXJ2(IJ)**2+UYJ2(IJ)**2+UZJ2(IJ)**2)
C5587    UXJ2(IJ)=UXJ2(IJ)/RR
C5597    UYJ2(IJ)=UYJ2(IJ)/RR
C5607    UZJ2(IJ)=UZJ2(IJ)/RR
C5617 95 CONTINUE
C5627    RETURN
C5637 980 WRITE(6,981)
C5647 981 FORMAT(/, ' **** ERROR IN WIRE INPUT')
C5657    STOP
C5667 990 WRITE(6,991)
C5677 991 FORMAT(/, ' **** ERROR IN JUNCTION INPUT')
C5687    STOP
C5697    END
C5707C *****
C5717C SUBROUTINE CSIMP(F,A,B,DEL,IMAX,S11,S,N,IER)
C5727C *****
C5737C COMPLEX SIMPSON INTEGRATION ROUTINE.
C5747C COMPLEX F,S11,S,SUMK
C5757    S11=(0.0,0.0)
C5767    S=(0.0,0.0)
C5777    N=0
C5787    BA=B-A
C5797    IF (BA)20,19,20
C5807 19 IER=1
C5817    RETURN
C5827 20 IF(DEL)22,22,23
C5837 22 IER=2
C5847    RETURN
C5857 23 IF(IMAX-1)24,24,25
C5867 24 IER=3
C5877    RETURN
C5887 25 X=BA/2,+A
C5897    NHALF=1
C5907
C5917

```

```

05927      SUMK=F(X)*BA*2./3.
05937      S=SUMK+(F(A)+F(B))*BA/6.
05947      DO 28 I=2,IMAX
05957      S11=S
05967      S=(S-SUMK/2.)/2.
05977      NHALF=NHALF*2
05987      ANHLF=NHALF
05997      FRSTX=A+(BA/ANHLF)/2.
06007      SUMK=F(FRSTX)
06017      XK=FRSTX
06027      KLAST=NHALF-1
06037      FINC=BA/ANHLF
06047      DO 26 K=1,KLAST
06057      XK=XK+FINC
06067      26 SUMK=SUMK+F(XK)
06077      SUMK=SUMK*2.+BA/(3.*ANHLF)
06087      S=S+SUMK
06097      IF(CABS(S).EQ.0.0) GO TO 29
06107      IF((CABS(S-S11)/CABS(S))-DEL) 29,28,28
06117      28 CONTINUE
06127      IER=4
06137      WRITE(6,1) S11,S
06147      1 FORMAT(//,29H INTEGRATION DID NOT CONVERGE,//
06157      1 50H THE PREVIOUS AND FINAL VALUES OF THE INTEGRAL ARE,
06167      2 13H RESPECTIVELY,4E12.3)
06177      GO TO 30
06187      29 IER=C
06197      30 N=2*NHALF
06207      RETURN
06217      END
06227C *****
06237      COMPLEX FUNCTION FUNC1(E)
06247C *****
06257      COMPLEX U
06267      COMMON /WAVE/ BK
06277      COMMON /INT/ M,RHO2,ZP
06287      DATA PI,U /3.14159265,(0.,1.)/
06297      R=SQRT(RHO2+(E-ZP)**2)
06307      FUNC1=CEXP(-U*BK*R)/R
06317      RETURN
06327      END
06337C *****
06347      COMPLEX FUNCTION FUNC2(E)
06357C *****
06367      COMPLEX U
06377      COMMON /WAVE/ BK
06387      COMMON /BOT2/ NP,RL
06397      COMMON /INT/ M,RHO2,ZP
06407      DATA PI,U /3.14159265,(0.,1.)/
06417      R=SQRT(RHO2+(E-ZP)**2)
06427      FUNC2=CEXP(-U*(N*PI*E/BL+BK*R))/R

```

```

06437      RETURN
06447      END
06457C *****
06467      SUBROUTINE PLOTA(X,Y,N,NR)
06477C *****
06487C
06497C WRITTEN 2/14/74   BY J. M. PUTNAM   DEPT 220   X23877
06507C
06517C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
06527C N IS THE NUMBER OF POINTS TO BE PLOTTED.
06537C NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
06547C NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
06557C NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
06567C
06577      REAL X(1),Y(1),HEAD(10)
06587      INTEGER LINE(101),BLANK,STAR,PLUS
06597      DATA BLANK,STAR,PLUS /1H,1H*,1H+/
06607      NC=51
06617      N10=(NC-1)/10
06627      WRITE(6,500)
06637 500  FORMAT(/,44H BODY COORDINATES, + INDICATES TRIANGLE PEAK)
06647      WRITE(6,504)
06657      XMIN=X(1)
06667      XMAX=X(1)
06677      YMIN=Y(1)
06687      YMAX=Y(1)
06697      DO 6 I=1,N
06707      IF(X(I).LT.XMIN) XMIN=X(I)
06717      IF(X(I).GT.XMAX) XMAX=X(I)
06727      IF(Y(I).LT.YMIN) YMIN=Y(I)
06737      IF(Y(I).GT.YMAX) YMAX=Y(I)
06747 6  CONTINUE
06757      DEL=XMAX-XMIN
06767      IF(YMAX-YMIN.GT.DEL) DEL=YMAX-YMIN
06777      XMAX=XMIN+DEL
06787      YMAX=YMIN+DEL
06797      DO 5 I=1,N10
06807      Z=1
06817 5  HEAD(I)=(XMAX-XMIN)*Z/N10+XMIN
06827      DY=(YMAX-YMIN)/(NR-1)
06837      IEDGE=1
06847      IF(X(1).EQ.X(N) .AND. Y(1).EQ.Y(N)) IEDGE=0
06857      Z=YMAX+DY
06867      YL=Z-DY/2.
06877      DO 7 J=1,NR
06887      DO 8 K=1,NC
06897 8  LINE(K)=BLANK
06907      Z=Z-DY
06917      YU=YL
06927      YL=Z-DY/2.
06937      DO 9 I=1,N
06947      IF(Y(I).GE.YU) GO TO 9

```

```

C6957      IF(Y(I).LT.YL) GO TO 9
C6967      K=(X(I)-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
C6977      IF(K.GT.NC) K=NC
C6987      LINE(K)=STAR
C6997      IF(MCD(I,2).EQ.1) LINE(K)=PLUS
C7007      IF(IEDGE.EQ.0) GO TO 9
C7017      IF(I.EQ.1 .OR. I.EQ.N) LINE(K)=STAR
C7027  9    CONTINUE
C7037      WRITE(6,508) Z,(LINE(K),K=1,NC)
C7047  7    CONTINUE
C7057      WRITE(6,504)
C7067      WRITE(6,3002)
C7077      WRITE(6,507) XMIN,(HEAD(I),I=1,N10)
C7087      RETURN
C7097  504  FORMAT (1X,14(1H-),14.,10(5H----.),1H- )
C7107  507  FORMAT(10X,11(F10.4))
C7117  508  FORMAT(1X,F12.4,1X,1HI,51A1,1HI )
C7127  3002 FORMAT(4X,7HYH / XH,4X,1HI,5(9X,1HI))
C7137      END
-----
C7140C
C7150      SUBROUTINE ZLIST(NI,NJ,LI,LJ,NAMI,NAMJ,Z)
C7160C  WRITE 7 TO THE LINE-PRINTER BY SUBMATRIX BY COLUMNS.
C7170      COMPLEX Z(1)
C7180      DIMENSION LI(1),LJ(1),NAMI(1),NAMJ(1)
C7190      NCOL=C
C7200      DO 100 I=1,NI
C7210  100  NCOL=NCOL+LI(I)
C7220      NR=1
C7230      DO 300 J=1,NJ
C7240      IF(LJ(J).EQ.0) GO TO 300
C7250      NC=1
C7260      DO 200 I=1,NI
C7270      IF(LI(I).EQ.C) GO TO 200
C7280      WRITE(6,1) NAMI(I),NAMJ(J)
C7290  1    FORMAT(/,2H Z,1X,A4,3H - ,A4)
C7300      KI=LI(I)
C7310      KJ=LJ(J)
C7320      K1=(NR-1)*NCOL+NC
C7330      DO 150 KK=1,KJ
C7340      K2=K1+KI-1
C7350      WRITE(6,2) (Z(K),K=K1,K2)
C7360  2    FORMAT(11,10G11.4)
C7370  150  K1=K1+NCOL
C7380      NC=NC+LI(I)
C7390      CONTINUE
C7400      NR=NR+LJ(J)
C7410  300  CONTINUE
C7420      RETURN
C7430      END
C7440C/END

```

PRECEDING PAGE BLANK NOT FILMED

```

CC100BOTZ,P30,T100,C4120000,STCK3.
CC110ACCOUNT,M03265,BOTZ4.
CC120BANNERS(OUTPUT)*J. PUTNAM*DEPT 220*BLD 110-4**
CC130DEFINE(OUTFIL=ZWWX)
CC138FTN(R=C,OPT=2)
CC140LDSET(PRESET=INDEF)
CC15CLGO.
CC21CEXIT.
CC230/FOR
CC240      PROGRAM BOTZWW(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
CC250      1 OUTFIL,TAPE7=OUTFIL)
CC271C
CC272C WIRE WIRE-JUNCTION INTERACTION MATRIX ELEMENTS
CC273C JUNCTION JUNCTION INTERACTION MATRIX ELEMENTS
CC274C RELEVANCE: STRONG COUPLING WIRE-BOR ANALYSIS
CC275C J.F.SHAFFER
CC276C MCDONNELL DOUGLAS RESEARCH LABORATORIES
CC277C SAINT LOUIS, MISSOURI
CC280C
CC290C JUNCTION MODELED AS HALF TRIANGLE FUNCTION, 1ST PT IS THE MAXIMUM
CC300C
CC302C COMPLEX Z(36C0)
CC305C COMMON /WAVE/ RK
CC310C COMMON /WIRE1/ NPW,X(110,3)
CC312C COMMON /WIRE2/ WL(110)
CC314C COMMON /WIRE3/ NW,INDW(6),RADW(5)
CC316C COMMON /WIRE4/ TWIRE(110,4),TPW(110,4),UW(110,3)
CC318C COMMON /JUNC1/ NJ,PADJ(10),RAND(10)
CC320C COMMON /JUNC2/ TJUNC(10,2),TPJ(10,2)
CC322C COMMON /JUNC3/ XJ(10,3,3),WLJ(10,2)
CC324C COMMON /JUNC4/ UJ(10,2,3)
CC326C COMMON /JUNC5/ UPT(10,3)
CC328C COMMON /JUNC6/ URZ(10,3)
CC330C COMMON /CONST/ PI,ETA,IMAG
CC335C COMPLEX IMAG,PSI
CC340C COMPLEX EJKP
CC345C DIMENSION SPV(3),FPV(3),RD(3)
CC350C DIMENSION U(3),XJJ(3),XIT(3)
CC355C INTEGER P,0
CC360C DIMENSION LI(2),LJ(2),NAMT(2),NAMJ(2)
CC365C DATA NAMI,NAMJ /4HWIRE,4HJUNC,4HWIRE,4HJUNC/
CC370C DATA PI,PERM,E0/3.1415927,1.2566E-6,8.85E-12/
CC375C DATA C,ETA/2.997925E8,377.0/
CC380C
CC385C IMAG=(0,C,1,C)
CC390C READ(5,1) RK
CC395C FORMAT(E15.7)
CC400 1 WRITE(6,2) RK
CC405 2 FORMAT(9H1      RK,/,E15.7)
CC410C
CC415C CALL DATAIN
CC420C
CC425C
CC430C

```

```

00535      WRITE(7) NW,NPW,NJ
00540      W=C*BK
00550C     COMPUTE WIRE JUNCTION MATRIX ELEMENTS
00560      NWIRE=(NPW-NW*3)/2
00565      IF(NJ.EQ.0) GO TO 167
00570C     DO 10 IJUNC=1,NJ
00580      ISTOP=0
00590C     DO 10 KWIRE=1,NW
00600      ISTART=ISTOP+1
00610      ISTOP=(INDW(KWIRE+1)-INDW(KWIRE)-3)/2 + ISTOP
00620C     DO 10 IWIRE=ISTART,ISTOP
00630      L1=(NWIRE+IJUNC-1)*(NWIRE+NJ)+IWIRE
00640      L1=(IWIRE-1)*(NWIRE+NJ)+NWIRE+IJUNC
00650C     Z(L1)=0.0
00660      KSM=2*(IWIRE-1) + 3*(KWIRE-1)
00670C     WIRE-SURFACE JUNCTION IMPEDANCE
00680C     SURFACE DISK RADIUS, R
00690      B=RADD(IJUNC)
00700      F1=BK*ETA/(4.*PI*2.*PI*B)
00710      INCR=2
00720      INCA=4
00730      DR=B/INCR
00740C     DALPHA=2.*PI/INCA
00750C     DO 11 P=1,4
00760      KS=KSM+P
00770C     DO 11 IR=1,INCR
00780      R=(IR-1)*DR+DR/2.
00790C     DO 11 IA=1,INCA
00800      ALPHA=(IA-1)*DALPHA + DALPHA/2.
00810C     SA=SIN(ALPHA)
00820      CA=CCS(ALPHA)
00830C     COMPUTE SURFACE DISK UNIT VECTOR, SOURCE POINT COORDINATES,
00840      DOT PRODUCT R.L, SOURCE-FIELD DISTANCE
00850C     DOT1=0.0
00860      RR=0.0
00870      DO 12 I=1,3
00880      RD(I)=CA*URZ(IJUNC,I) + SA*URZ(IJUNC,I)
00890      DOT1=DOT1 + RD(I)*UW(KS,I)
00900

```

```

01000 SPV(I)=XJ(IJUNC,1,I) + P*RD(I)
01010 XR=SPV(I) - 0.5*(X(KS+1,I)+X(KS,I))
01020 12 RR=RR + XR*XR
01030 RR=SQRT(RR)
01040C ARG=BK*RR
01048 EJKR=(COS(ARG) - IMAG*SIN(ARG)) / RR
01050C FACT=F1*WL(KS)*DR*DALPHA
01060C Z(L1)=Z(L1) + (TWIRE(IWIRE,P)*(R-B)*DOT1
01070C 1 -TPW(IWIRE,P)/BK/BK)*IMAG*EJKR*FACT
01080C 11 CONTINUE
01090C
01100C WIRE-WIRE JUNCTION IMPEDANCE
01110C DO 5 P=1,4
01120C KS=KSW+P
01130C DO 5 Q=1,2
01140C DOT=C.0
01150C DO 4 K=1,3
01160C DOT=DOT+UM(KS,K)*UJ(IJUNC,Q,K)
01170C XJJ(K)=0.5*(XJ(IJUNC,Q+1,K) + XJ(IJUNC,Q,K))
01180C XII(K)=0.5*(X(KS+1,K) + X(KS,K))
01190C 4 U(K)=UM(KS,K)
01200C DLJ=WL(KS)
01210C CALL GREEN(PSI,XJJ,XII,U,DLJ,BK,RADW(KWIRE))
01220C Z(L1)=Z(L1) + (PERM*W*DOT*TWIRE(IWIRE,P)
01230C 1*IJUNC(IJUNC,Q) - TPW(IWIRE,P)*TPJ(IJUNC,Q)/W/E0)
01240C 2*IMAG*PSI*WLJ(IJUNC,Q)
01250C 5 CONTINUE
01260C Z(L1)=Z(L1)
01270C 10 CONTINUE
01280C
01290C COMPUTE JUNCTION-JUNCTION MATRIX ELEMENTS
01300C DO 100 IJUNC=1,NJ
01310C DO 100 JJUNC=1,NJ
01320C I=IJUNC
01330C J=JJUNC
01340C L1=(NWIRE+JJUNC-1)*(NWIRE+NJ)+NWIRE+IJUNC
01350C Z(L1)=C.C
01360C DO 19 Q=1,2
01370C DO 19 P=1,2
01380C

```

```

01510      DOT=C.0
01520      DO 18 K=1,3
01530      DOT=DOT + UJ(I,P,K)*UJ(J,Q,K)
01540      XJJ(K)=0.5*(XJ(J,Q+1,K) + XJ(J,Q,K))
01550      XII(K)=0.5*(XJ(I,P+1,K) + XJ(I,P,K))
01560 18    U(K)=UJ(I,P,K)
01570      DLJ=WLJ(I,P)
01580C
01590C      CALL GREEN(PSI,XJJ,XII,U,DLJ,BK,RADJ(I))
01600C
01610      Z(L1)=Z(L1) + (PERM*W*DOT+TJUNC(I,P)+TJUNC(J,Q)
01620      1-TPJ(I,P)*TPJ(J,Q)/W/EO) * IMAG*PSI*WLJ(J,Q)
01630 19    CONTINUE
01640 20    CONTINUE
01650C
01660C      HALF TRIANGLE-SURFACE DISK IMPEDANCE
01670C
01680C      JJUNC--HALF TRIANGLE
01690C
01700C      IJUNC--SURFACE DISK
01710C
01720C      IF(IJUNC.EQ.JJUNC) GO TO 115
01730C
01731C      MJUNC=IJUNC
01732C      NJUNC=JJUNC
01733C      DO 110 M=1,2
01734C      IF(M.EQ.2) MJUNC=JJUNC
01735C      IF(M.EQ.2) NJUNC=IJUNC
01736C      A=RADJ(MJUNC)
01740C
01750C      SURFACE DISK RADIUS, B
01760C      B=RADD(MJUNC)
01770C      F1=BK*ETA/(4.*PI*2.*PI*(B-A))
01780C      INCR=2
01790C      INCA=4
01800C      OR=(B-A)/INCR
01810C      DALPHA=2.*PI/INCA
01820C
01830C      DO 110 Q=1,2
01840C      DO 110 IR=1,INCR
01850C      R=A + (IR-0.5)*OR
01860C
01870C      DO 110 IA=1,INCA
01880C      ALPHA=(IA-1)*DALPHA + DALPHA/2.
01890C      SA=SIN(ALPHA)
01900C      CA=CCS(ALPHA)
01910C
01920C      COMPUTE SURFACE DISK UNIT VECTOR, SOURCE POINT COORDINATES,
01930C      DOT PRODUCT R.L, SOURCE-FIELD DISTANCE
01940C
01950C      DOT1=C.C
01960C      RR=0.0
01962

```

```

01970      DD 120 I=1,3
01980      RD(I)=CA*URT(MJUNC,I) + SA*URZ(MJUNC,I)
01990      DOT1=DOT1 + RD(I)*UJ(NJUN,Q,I)
02000      SPV(I)=XJ(MJUNC,1,I) + R*RD(I)
02010      XR=SPV(I) - 0.5*(XJ(NJUN,Q+1,I)+XJ(NJUN,Q,I))
02020 120  RR=RR + XR*XR
02030      RR=SQRT(RR)
02040C
02048      ARG=BK*RR
02050      EJKR=(COS(ARG) - IMAG*SIN(ARG)) / RR
02060C
02070      FACT=F1*WLJ(NJUN,Q)*DR*DALPHA
02080C
02090      Z(L1)=Z(L1) + (TJUNC(NJUN,Q)*(R-B)*DOT1
02100 110  -TPJ(NJUN,Q)/BK/BK)*IMAG+EJKR*FACT
02104      GO TO 116
02106 115  CONTINUE
02108C
02110C      SELF TERM FOR WIRE-DISK (MULT BY 2)
02112C
02114      B=RADD(IJUNC)
02115      A=RADD(IJUNC)
02116      HC=WLJ(IJUNC,1)+WLJ(IJUNC,2)
02117      T3=0.0
02118      INCR=40
02119      DR=(B-A)/INCR
02120      DD 521 IR=1,INCR
02121      R=A+(IR-0.5)*DR
02122      T3=T3+DR*ALOG((HO+SQR(HO*HO+R*R))/R)
02124 521  CONTINUE
02126      T3=T3/(B-A)/HO
02128      Z(L1)=Z(L1) + 2.*(ETA/4./PI +
02133 1  IMAG*ETA/4./PI/RK*T3)
02134C
02135 116  CONTINUE
02140C      DISK-DISK INTERACTION
02150C
02160C      SURFACE DISK RADIUS
02170      B1=RADD(IJUNC)
02180      B2=RADD(JJUNC)
02190      IF(IJUNC.NE.JJUNC) GO TO 135
02200C
02210C      SELF TERM FOR DISK-DISK
02220C
02225      INCB=40
02227      INCR=16
02230      B=B1
02232      A=RADD(IJUNC)
02235C
02240      DR=(B-A)/INCR
02242      DBETA=PI/INCB

```

```

C2245C
C2246C
C2247C
C2248C
C2249C
C2250C
C2251C
C2252C
C2253C
C2254C
C2255C
C2256C
C2257C
C2258C
C2259C
C2260C
C2261C
C2262C
C2263C
C2264C
C2265C
C2266C
C2267C
C2268C
C2269C
C2270C
C2271C
C2272C
C2273C
C2274C
C2275C
C2276C
C2277C
C2278C
C2279C
C2280C
C2281C
C2282C
C2283C
C2284C
C2285C
C2286C
C2287C
C2288C
C2289C
C2290C
C2291C
C2292C
C2293C
C2294C
C2295C
C2296C
C2297C
C2298C
C2299C
C2300C
C2301C
C2302C
C2303C
C2304C
C2305C
C2306C
C2307C
C2308C
C2309C
C2310C
C2311C
C2312C
C2313C
C2314C
C2315C
C2316C
C2317C
C2318C
C2319C
C2320C
C2321C
C2322C
C2323C
C2324C
C2325C
C2326C
C2327C
C2328C
C2329C
C2330C
C2331C
C2332C
C2333C
C2334C
C2335C
C2336C
C2337C
C2338C
C2339C
C2340C
C2341C
C2342C
C2343C
C2344C
C2345C
C2346C
C2347C
C2348C
C2349C
C2350C
C2351C
C2352C
C2353C
C2354C
C2355C
C2356C
C2357C
C2358C
C2359C
C2360C
C2361C
C2362C
C2363C
C2364C
C2365C
C2366C
C2367C
C2368C
C2369C
C2370C
C2371C
C2372C
C2373C
C2374C
C2375C
C2376C
C2377C
C2378C
C2379C
C2380C
C2381C
C2382C
C2383C
C2384C
C2385C
C2386C
C2387C
C2388C
C2389C
C2390C
C2391C
C2392C
C2393C
C2394C
C2395C
C2396C
C2397C
C2398C
C2399C
C2400C
C2401C
C2402C
C2403C
C2404C
C2405C
C2406C
C2407C
C2408C
C2409C
C2410C
C2411C
C2412C
C2413C
C2414C
C2415C
C2416C
C2417C
C2418C
C2419C
C2420C
C2421C
C2422C
C2423C
C2424C
C2425C
C2426C
C2427C
C2428C
C2429C
C2430C
C2431C
C2432C
C2433C
C2434C
C2435C
C2436C
C2437C
C2438C
C2439C
C2440C
C2441C
C2442C
C2443C
C2444C
C2445C
C2446C
C2447C
C2448C
C2449C
C2450C
C2451C
C2452C
C2453C
C2454C
C2455C
C2456C
C2457C
C2458C
C2459C
C2460C
C2461C
C2462C
C2463C
C2464C
C2465C
C2466C
C2467C
C2468C
C2469C
C2470C
C2471C
C2472C
C2473C
C2474C
C2475C
C2476C
C2477C
C2478C
C2479C
C2480C
C2481C
C2482C
C2483C
C2484C
C2485C
C2486C
C2487C
C2488C
C2489C
C2490C
C2491C
C2492C
C2493C
C2494C
C2495C
C2496C
C2497C
C2498C
C2499C
C2500C
C2501C
C2502C
C2503C
C2504C
C2505C
C2506C
C2507C
C2508C
C2509C
C2510C
C2511C
C2512C
C2513C
C2514C
C2515C
C2516C
C2517C
C2518C
C2519C
C2520C
C2521C
C2522C
C2523C
C2524C
C2525C
C2526C
C2527C
C2528C
C2529C
C2530C
C2531C
C2532C
C2533C
C2534C
C2535C
C2536C
C2537C
C2538C
C2539C
C2540C
C2541C
C2542C
C2543C
C2544C
C2545C
C2546C
C2547C
C2548C
C2549C
C2550C
C2551C
C2552C
C2553C
C2554C
C2555C
C2556C
C2557C
C2558C
C2559C
C2560C
C2561C
C2562C
C2563C
C2564C
C2565C
C2566C
C2567C
C2568C
C2569C
C2570C
C2571C
C2572C
C2573C
C2574C
C2575C
C2576C
C2577C
C2578C
C2579C
C2580C
C2581C
C2582C
C2583C
C2584C
C2585C
C2586C
C2587C
C2588C
C2589C
C2590C
C2591C
C2592C
C2593C
C2594C
C2595C
C2596C
C2597C
C2598C
C2599C
C2600C
C2601C
C2602C
C2603C
C2604C
C2605C
C2606C
C2607C
C2608C
C2609C
C2610C
C2611C
C2612C
C2613C
C2614C
C2615C
C2616C
C2617C
C2618C
C2619C
C2620C
C2621C
C2622C
C2623C
C2624C
C2625C
C2626C
C2627C
C2628C
C2629C
C2630C
C2631C
C2632C
C2633C
C2634C
C2635C
C2636C
C2637C
C2638C
C2639C
C2640C
C2641C
C2642C
C2643C
C2644C
C2645C
C2646C
C2647C
C2648C
C2649C
C2650C
C2651C
C2652C
C2653C
C2654C
C2655C
C2656C
C2657C
C2658C
C2659C
C2660C
C2661C
C2662C
C2663C
C2664C
C2665C
C2666C
C2667C
C2668C
C2669C
C2670C
C2671C
C2672C
C2673C
C2674C
C2675C
C2676C
C2677C
C2678C
C2679C
C2680C
C2681C
C2682C
C2683C
C2684C
C2685C
C2686C
C2687C
C2688C
C2689C
C2690C
C2691C
C2692C
C2693C
C2694C
C2695C
C2696C
C2697C
C2698C
C2699C
C2700C
C2701C
C2702C
C2703C
C2704C
C2705C
C2706C
C2707C
C2708C
C2709C
C2710C
C2711C
C2712C
C2713C
C2714C
C2715C
C2716C
C2717C
C2718C
C2719C
C2720C
C2721C
C2722C
C2723C
C2724C
C2725C
C2726C
C2727C
C2728C
C2729C
C2730C
C2731C
C2732C
C2733C
C2734C
C2735C
C2736C
C2737C
C2738C
C2739C
C2740C
C2741C
C2742C
C2743C
C2744C
C2745C
C2746C
C2747C
C2748C
C2749C
C2750C
C2751C
C2752C
C2753C
C2754C
C2755C
C2756C
C2757C
C2758C
C2759C
C2760C
C2761C
C2762C
C2763C
C2764C
C2765C
C2766C
C2767C
C2768C
C2769C
C2770C
C2771C
C2772C
C2773C
C2774C
C2775C
C2776C
C2777C
C2778C
C2779C
C2780C
C2781C
C2782C
C2783C
C2784C
C2785C
C2786C
C2787C
C2788C
C2789C
C2790C
C2791C
C2792C
C2793C
C2794C
C2795C
C2796C
C2797C
C2798C
C2799C
C2800C
C2801C
C2802C
C2803C
C2804C
C2805C
C2806C
C2807C
C2808C
C2809C
C2810C
C2811C
C2812C
C2813C
C2814C
C2815C
C2816C
C2817C
C2818C
C2819C
C2820C
C2821C
C2822C
C2823C
C2824C
C2825C
C2826C
C2827C
C2828C
C2829C
C2830C
C2831C
C2832C
C2833C
C2834C
C2835C
C2836C
C2837C
C2838C
C2839C
C2840C
C2841C
C2842C
C2843C
C2844C
C2845C
C2846C
C2847C
C2848C
C2849C
C2850C
C2851C
C2852C
C2853C
C2854C
C2855C
C2856C
C2857C
C2858C
C2859C
C2860C
C2861C
C2862C
C2863C
C2864C
C2865C
C2866C
C2867C
C2868C
C2869C
C2870C
C2871C
C2872C
C2873C
C2874C
C2875C
C2876C
C2877C
C2878C
C2879C
C2880C
C2881C
C2882C
C2883C
C2884C
C2885C
C2886C
C2887C
C2888C
C2889C
C2890C
C2891C
C2892C
C2893C
C2894C
C2895C
C2896C
C2897C
C2898C
C2899C
C2900C
C2901C
C2902C
C2903C
C2904C
C2905C
C2906C
C2907C
C2908C
C2909C
C2910C
C2911C
C2912C
C2913C
C2914C
C2915C
C2916C
C2917C
C2918C
C2919C
C2920C
C2921C
C2922C
C2923C
C2924C
C2925C
C2926C
C2927C
C2928C
C2929C
C2930C
C2931C
C2932C
C2933C
C2934C
C2935C
C2936C
C2937C
C2938C
C2939C
C2940C
C2941C
C2942C
C2943C
C2944C
C2945C
C2946C
C2947C
C2948C
C2949C
C2950C
C2951C
C2952C
C2953C
C2954C
C2955C
C2956C
C2957C
C2958C
C2959C
C2960C
C2961C
C2962C
C2963C
C2964C
C2965C
C2966C
C2967C
C2968C
C2969C
C2970C
C2971C
C2972C
C2973C
C2974C
C2975C
C2976C
C2977C
C2978C
C2979C
C2980C
C2981C
C2982C
C2983C
C2984C
C2985C
C2986C
C2987C
C2988C
C2989C
C2990C
C2991C
C2992C
C2993C
C2994C
C2995C
C2996C
C2997C
C2998C
C2999C
C3000C

```

A2=A*A
B2=B*B
F1=BK*ETA/(2.*PI*(B-A))*2
F=0.0
DO 222 IR=1, INCR
BETA=(IR-0.5)*DBETA
CB=COS(BETA)
DO 222 IR=1, INCR
R=A + (IR-0.5)*OR
R2=R*R
RCB=R*CB
T1=SQRT(R2 - 2.*R*RCB + R2)
T2=SQRT(A2 - 2.*A*RCB + R2)
FI1=ALCG((T1 + B - RCB) / (T2 + A - RCB))
FI2=T1 - T2 + RCB*FI1
F=F + ((B*(B-R)*CB - 1./BK/BK)*FI1 - (B-R)*CB*FI2)*DR*DBETA
222 CONTINUE
Z(L1)=Z(L1) + F1*IMAG*(PI*IMAG*(R-A))*2/BK
1 + F
GD TC 131
135 CONTINUE
F1=BK*ETA/(4.*PI)/(2.*PI))*2/B1/B2
INCR=2
INCA=4
DR1=B1/INCR
DR2=B2/INCR
DALPHA=2.*PI/INCA
DO 130 IR1=1, INCR
R1=(IR1-0.5)*DR1
DO 130 IR2=1, INCR
R2=(IR2-0.5)*DR2
DO 130 IA1=1, INCA
A1=(IA1-0.5)*DALPHA
SA1=SIN(A1)
CA1=COS(A1)
DO 130 IA2=1, INCA
A2=(IA2-0.5)*DALPHA
SA2=SIN(A2)
CA2=COS(A2)
COMPUTE SURFACE DISK UNIT VECTORS; SOURCE POINT-FIELD POINT

```

G271CC COORDINATES & DISTANCE; DOT PRODUCT R1*R2
G2720C
G2730C DOT1=C.C
G2732C RR=0.0
G2740C DO 14C I=1,3
G2750C RD1=CA1*URT(IJUNC,I) + SA1*URZ(IJUNC,I)
G2760C RD2=CA2*URT(IJUNC,I) + SA2*URZ(IJUNC,I)
G2770C DOT1=DOT1 + RD1*RD2
G2780C SPV(I)=XJ(IJUNC,1,I) + P1*RD1
G2790C FPV(I)=XJ(IJUNC,1,I) + P2*RD2
G2800C XR=SPV(I) - FPV(I)
G2810C 14C RR=RR + XR*XR
G2815C RR=RR+XR
G2820C
G2828C ARG=BK*RR
G2830C EJKR=(COS(ARG) - IMAG*SIN(ARG)) / RR
G2840C
G2850C FACT=F1*DALPHA*DALPHA*DR1*DR2
G2860C
G2870C Z(L1)=Z(L1) + ((R1-R1)*(R2-R2)*DOT1
G2880C 1 -1./BK/BK)*IMAG*EJKR*FACT
G2900C 130 CONTINUE
G2910C 131 CONTINUE
G2920C
G2930C 100 CONTINUE
G2935C 167 CONTINUE
G2940C
G2950C COMPUTE WIRE-WIRE MATRIX ELEMENTS, ZWW
G2960C
G2970C ISTOP=0
G2980C
G2990C DO 30 KWIRE=1,NW
G3000C
G3010C ISTART=ISTOP + 1
G3020C ISTOP=(INDW(KWIRE+1)-INDW(KWIRE)-3)/2 + ISTOP
G3030C
G3040C DO 3C I=ISTART,ISTOP
G3050C
G3060C JSTOP=0
G3070C
G3080C DO 3C KJWIRE=1,NW
G3090C
G3100C JSTART=JSTOP+1
G3110C JSTOP=(INDW(KJWIRE+1)-INDW(KJWIRE)-3)/2 + JSTOP
G3120C
G3130C DO 3C J=JSTART,JSTOP
G3140C
G3150C L1=(J-1)*(NWIRE+NJ)+I
G3160C Z(L1)=0.0
G3170C IK=2*(I-1) + 3*(KWIRE-1)
G3180C JK=2*(J-1) + 3*(KJWIRE-1)

```

```

033100      DD 29 P=1,4
033200      IP=IX,P
033210C
033220      DD 29 Q=1,4
033230      JB=JK+Q
033240      DOT=0.0
033250C
033260      DD 28 K=1,3
033270      DOT=DOT+UW(IP,K)+UW(JQ,K)
033280      XJJ(K)=0.5*(X(IP+1,K)+X(IP,K))
033290      XII(K)=0.5*(X(JQ+1,K)+X(JQ,K))
033300      U(K)=UW(IP,K)
033310      DLJ=WL(IP)
033320C
033330      CALL GREEN(P,SI,XJJ,XII,U,DLJ,BK,RADW(KWIRE))
033340C
033350      Z(L1)=Z(L1) + (PERM*W*DOT*TWIRE(I,P)*TWIRE(J,Q) -
033360      1 TPW(I,P)*TPW(J,Q)/W/EO) +
033370      2 IMAG*PSI*WL(JQ)
033380C
033390      29 CONTINUE
033400      30 CONTINUE
033410C PRINT OUT IMPEDANCE MATRIX Z BY SUBMATRIX BY COLUMNS.
033420      LI(1)=NWIRE
033430      LI(2)=NJ
033440      LJ(1)=NWIRE
033450      LJ(2)=NJ
033460C CALL ZLIST(2,2,LI,LJ,NAMI,NAMJ,Z)
033470      WRITE(7) TO DISK
033480      LNW=(NWIRE+NJ)**2
033490      WRITE(7) (Z(I),I=1,LNW)
033500      STOP
033510C *****
033520      SUBROUTINE GREEN(P,SI,XJ,XI,U,DLJ,BK,RADW)
033530C
033540      COMMON/ CONST / PI,ETA,IMAG
033550      COMPLEX PSI,IMAG
033560      DIMENSION XJ(1),XI(1),U(1)
033570C
033580C COMPUTE DISTANCE BETWEEN CENTER OF SEGMENTS XJ & XI
033590      WR=RADW*RADW
033600      RQ2=C.0
033610      ZJ=0.0
033620      DD 10 K=1,3
033630      SD=XJ(K)-XI(K)
033640      RQ2=RQ2+SD*SD
033650      10 ZJ=ZJ+SD*U(K)
033660C
033670      ZJ2=ZJ*ZJ
033680      RHO2=RQ2 - ZJ2
033690      ZJ=SQRT(ZJ2)
033700

```

```

C389CC      IF(RHO2.LT.WR) RHO2=WR
C390CC      RO=SQRT(RO2)
C392CC      RHO=SQRT(RHO2)
C393CC      ARG=BK*RO
C394CC      ALPHA=DLJ/2.
C395CC
C396CC      IF(RO.GE.(1C.*ALPHA)) GO TO 2C
C397CC
C398CC      ZP=ZJ+ALPHA
C398CC      ZM=ZJ-ALPHA
C399CC      ZRP=SQRT(RHO2+ZP*ZP)
C399CC      ZRM=SQRT(RHO2+ZM*ZM)
C400CC      IF(ZM.LE.0.0) GO TO 12
C401CC      TERM=(ZP*ZRP)/(ZM*ZRM)
C402CC      A11=ALOG(TERM)
C403CC      GO TO 13
C404CC      12 A11=ALOG((ZP*ZRP)*(-ZM*ZRM)/RHO2)
C405CC      13 A12=2.*ALPHA
C406CC      A13=(ZP*ZRP - ZM*ZRM + RHO2*A11)/2.0
C407CC      A14=A12*RHO2 + (A12*ALPHA*ALPHA + 3.*A12*ZJ2)/3.0
C408CC      BK2=BK*BK
C408CC      BK3=BK2*BK
C409CC      PSI=(COS(ARG) - IMAG*SIN(ARG))/(4.*PI)
C410CC      PSI=PSI*(A11 - IMAG*BK*(A12-RO*A11) - BK2*(A13-2.*RO*A12
C411CC      1 +RO2*A11)/2. + IMAG*BK3*(A14-3.*RO*A13+3.*RO2*A12
C412CC      2 -RO*RO2*A11)/6.
C413CC      RETURN
C414CC
C415CC      20 CONTINUE
C416CC
C417CC      ZR=ZJ/RO
C417CC      ZR2=ZR*ZR
C417CC      ZR4=ZR2*ZR2
C418CC      AR=ALPHA/RO
C418CC      AR2=AR*AR
C418CC      AR3=AR2*AR
C418CC      AR4=AR3*AR
C419CC      AK=BK*ALPHA
C419CC      AK2=AK*AK
C419CC      AK3=AK2*AK
C419CC      AK4=AK3*AK
C420CC      AC=1.+AR2/6.*(3.*ZR2-1.0) + AR4/4C.*(3.-30.*ZR2+35.*ZR4)
C421CC      A1=AR/6.*(3.*ZR2-1.0) + AR3/40.*(3.-30.*ZR2+35.*ZR4)
C422CC      A2=-ZR2/6. - AR2/40.*(1.-12.*ZR2+15.*ZR4)
C423CC      A3=AR/60.*(3.*ZR2-5.*ZR4)
C424CC      A4=ZR4/120.
C425CC      PSI=(COS(ARG) - IMAG*SIN(ARG))/(4.*PI*RO) + (2.*ALPHA)
C426CC      PSI=PSI*(AO + IMAG*AK*A1 + AK2*A2 + IMAG*AK3*A3 + AK4*A4)
C427CC      RETURN
C428CC
C429CC      END

```

```

04540C -----
04550 SUBROUTINE DATIN
04560C
04570 COMMON /WIRE1/ NPW,X(110,3)
04572 COMMON /WIRE2/ WL(110)
04574 COMMON /WIRE3/ NW,INDW(6),RADW(5)
04576 COMMON /WIRE4/ TWIRE(110,4),TPW(110,4),UW(110,3)
04578 COMMON /JUNC1/ NJ,RADJ(10),RAOD(10)
04580 COMMON /JUNC2/ TJUNC(10,2),TPJ(10,2)
04582 COMMON /JUNC3/ XJ(10,3,3),WLJ(10,2)
04584 COMMON /JUNC4/ UJ(10,2,3)
04586 COMMON /JUNC5/ URT(10,3)
04588 COMMON /JUNC6/ URZ(10,3)
04635 DIMENSION INDJW(10),INDTJ(10),UXJ(10),UYJ(10),UZJ(10)
04640C
04650 DIMENSION DL(3)
04670C
04680 CALL SBTIN
04737C
04739 CALL SCAPIN
04740 READ(5,16) NW,NPW,NJ
04750 16 FORMAT(3I3)
04760C
04780C NPW=NO. WIRE POINTS
04790C NW= NO. WIRES
04800C
04810 WRITE(6,1) NW,NPW,NJ
04820 1 FORMAT(26H NW NPW NJ ,/,3I8)
04830C
04860C READ IN WIRE COORDINATES
04870 DO 17 I=1,3
04880 17 READ(5,12) (X(J,I),J=1,NPW)
04885 12 FORMAT(10F8,4)
04890C
04900C INDW(I)=START POINT FOR EACH WIRE
04910 READ(5,18) (INDW(I),I=1,NW)
04915 18 FORMAT(10I8)
04920 INDW(NW+1)=NPW + 1
04930C
04932 READ(5,12) (RADW(I),I=1,NW)
04934C READ JUNCTION COORDINATES
04936 IF(NJ.EQ.0) GO TO 50
04938 READ(5,18) (INDJW(I),I=1,NJ)
04940 READ(5,12) (RAOD(I),I=1,NJ)
04942 READ(5,12) (UXJ(I),I=1,NJ)
04944 READ(5,12) (UYJ(I),I=1,NJ)
04946 READ(5,12) (UZJ(I),I=1,NJ)
04948 50 IERW=0
04950 IERJ=0
04952 WRITE(6,61)
04954 61 FORMAT(/,22X,'WIRE COORDINATES',20X,' JUNCTION PARAMETERS',/,
04956 1 26X,'IW',4X,'XW',6X,'YW',6X,'ZW',

```

```

04958      2 10X,'IJ',4X,'RADD',5X,'UXJ',5X,'UYJ',5X,'UZJ')
04960      IJ=1
04962C THIS LOOP LISTS WIRE/JUNCTION POINTS, WHILE CHECKING THE FOLLOWING:
04964C 1) EACH WIRE MUST CONTAIN AN ODD NUMBER OF POINTS.
04966C 2) EACH JUNCTION MUST EITHER START OR TERMINATE A WIRE.
04968C 3) CHECK THAT ALL JUNCTION POINTS ARE FOUND.
04970      DO 100 IW=1,NW
04972      WRITE(6,62) IW,RADW(IW)
04974 62  FORMAT(2X,'WIRE',I3,' RADW=',F8.4)
04976      I1=INDW(IW)
04978      I2=INDW(IW+1)-1
04980C CHECK FOR AN ODD NUMBER OF POINTS ON WIRE IW.
04982      IF(MOD(I2-I1+1,2).NE.1) IERW=1
04984      DO 90 I=I1,I2
04986      WRITE(6,63) I,X(I,1),X(I,2),X(I,3)
04988 63  FORMAT(2X,I3,3F8.4)
04990C CHECK IF WIRE POINT I IS A JUNCTION POINT.
04992      IF(IJ.GT.NJ) GO TO 90
04994      IF(INDJW(IJ).NE.I) GO TO 90
04996C CHECK THAT JUNCTION POINT IJ IS AT THE START OR END OF WIRE.
04998      IF(I.NE.I1 .AND. I.NE.I2) IERJ=1
05000      WRITE(6,64) IJ,RADD(IJ),UXJ(IJ),UYJ(IJ),UZJ(IJ)
05002 64  FORMAT(1X,5X,I3,4F8.4)
05004C DETERMINE DIRECTION IN WHICH WIRE LEAVES JUNCTION POINT.
05006      INDTJ(IJ)=1
05008      IF(I.EQ.I2) INDTJ(IJ)=-1
05010C COMPUTE JUNCTION PARAMETERS.
05012      RADJ(IJ)=RADW(IW)
05014      DO 105 J=1,3
05016      XJ(IJ,1,J)=X(I,J)
05018      XJ(IJ,2,J)=X(I+INDTJ(IJ),J)
05020      XJ(IJ,3,J)=X(I+2*INDTJ(IJ),J)
05022 105 CONTINUE
05024      IJ=IJ+1
05026      90 CONTINUE
05028      100 CONTINUE
05030C CHECK FOR WIRE OR JUNCTION INPUT ERRORS.
05032      IF(IERW.NE.0) GO TO 980
05034      IF(IJ-1.NE.NJ .OR. IERJ.NE.0) GO TO 990
05036C WIRE PARAMETERS: UNIT VECTOR, UW(I) ; LENGTH, WL(NSEG)
05038      NSTOP=0
05040      DO 40 KWIRE=1,NW
05042      ISTART=INDW(KWIRE)
05044      ISTOP=INDW(KWIRE+1)-2
05046C      DO 35 NSEG=ISTART,ISTOP
05048      IP=NSEG+1
05050      DO 36 I=1,3
05052 36  OL(I)= X(IP,I)-X(NSEG,I)
05054C

```

```

C5520      WL(NSEG)=SQRT( DL(1)*DL(1)+DL(2)*DL(2)+DL(3)*DL(3) )
C5530C
C5540      DO 34 I=1,3
C5550 34    UW(NSEG,I)=DL(I)/WL(NSEG)
C5560C
C5580C
C5590 35    CONTINUE
C5600C    COMPUTE TRIANGLE FUNCTIONS FOR EACH WIRE
C5610      NSTART=NSTOP+1
C5620      NSTOP=NSTOP + (ISTOP-ISTART-1)/2
C5630      DO 38 N=NSTART,NSTOP
C5640      J1=2*(N-1)+1+3*(KWIRE-1)
C5650      J2=J1+1
C5660      J3=J2+1
C5670      J4=J3+1
C5680      DEL1=WL(J1)+WL(J2)
C5690      DEL2=WL(J3)+WL(J4)
C5700      TWIRE(N,1)=0.5*WL(J1)/DEL1
C5710      TWIRE(N,2)=(WL(J1)+0.5*WL(J2))/DEL1
C5720      TWIRE(N,3)=(WL(J4)+0.5*WL(J3))/DEL2
C5730      TWIRE(N,4)=0.5*WL(J4)/DEL2
C5740      TPW(N,1)=1./DEL1
C5750      TPW(N,2)=1./DEL1
C5760      TPW(N,3)=-1./DEL2
C5770      TPW(N,4)=-1./DEL2
C5780 38    CONTINUE
C5790 40    CONTINUE
C5800C
C5940      IF(NJ.EQ.0) RETURN
C5950C    COMPUTE JUNCTION UNIT VECTORS AND SEGMENT LENGTHS
C5960      DO 116 N=1,NJ
C5970      DO 116 J=1,2
C5980      DO 114 I=1,3
C5990 114    DL(I)=XJ(N,J,I)-XJ(N,J,I)
C6000      WLJ(N,J)=SQRT(DL(1)*DL(1)+DL(2)*DL(2)+DL(3)*DL(3))
C6010      DO 115 I=1,3
C6020 115    UJ(N,J,I)=DL(I)/WLJ(N,J)
C6030 116    CONTINUE
C6040C
C6050C    COMPUTE HALF TRIANGLE FUNCTION CENTROID VALUES
C6060      DO 120 N=1,NJ
C6070      DEL2=WLJ(N,1)+WLJ(N,2)
C6080      TPJ(N,1)=-1./DEL2
C6090      TPJ(N,2)=-1./DEL2
C6100      TJUNC(N,1)=(WLJ(N,2)+0.5*WLJ(N,1))/DEL2
C6110 120    TJUNC(N,2)=0.5*WLJ(N,2)/DEL2
C6120C    COMPUTE UNIT VECTORS FOR JUNCTION(DISK).
C6130      DO 95 IJ=1,NJ
C6140C    NORMAL UNIT VECTOR (UJ).
C6150      RR=SQRT(UXJ(IJ)**2+UYJ(IJ)**2+UZJ(IJ)**2)
C6160      UXJ(IJ)=UXJ(IJ)/RR
C6170      UYJ(IJ)=UYJ(IJ)/RR

```

```

06122      UZJ(IJ)=UZJ(IJ)/RR
06124C FIND 2 ORTHOGONAL UNIT VECTORS IN THE PLANE OF THE DISK (URT & URZ).
06126      URT(IJ,1)=0.0
06128      URT(IJ,2)=0.0
06130      URT(IJ,3)=0.0
06132      URZ(IJ,1)=0.0
06134      URZ(IJ,2)=0.0
06136      URZ(IJ,3)=0.0
06138      IF(UXJ(IJ).EQ.0.0) GO TO 91
06140C FIND INTERSECTION WITH X-Y PLANE.
06142      URT(IJ,2)=1.0
06144      URT(IJ,1)=-UYJ(IJ)/UXJ(IJ)
06146C FIND INTERSECTION WITH X-Z PLANE.
06148      URZ(IJ,3)=1.0
06150      URZ(IJ,1)=-UZJ(IJ)/UXJ(IJ)
06152      GO TO 94
06154 91 IF(UYJ(IJ).EQ.0.0) GO TO 92
06156C FIND INTERSECTION WITH Y-Z PLANE.
06158      URT(IJ,3)=1.0
06160      URT(IJ,2)=-UZJ(IJ)/UYJ(IJ)
06162C FIND INTERSECTION WITH X-Y PLANE.
06164      URZ(IJ,1)=1.0
06166      URZ(IJ,2)=-UXJ(IJ)/UYJ(IJ)
06168      GO TO 94
06170 92 IF(UZJ(IJ).EQ.0.0) GO TO 94
06172C FIND INTERSECTION WITH X-Z PLANE.
06174      URT(IJ,1)=1.0
06176      URT(IJ,3)=-UXJ(IJ)/UZJ(IJ)
06178C FIND INTERSECTION WITH Y-Z PLANE.
06180      URZ(IJ,2)=1.0
06182      URZ(IJ,3)=-UYJ(IJ)/UZJ(IJ)
06184 94 CONTINUE
06186      RR=SQRT(URT(IJ,1)**2+URT(IJ,2)**2+URT(IJ,3)**2)
06188      URT(IJ,1)=URT(IJ,1)/RR
06190      URT(IJ,2)=URT(IJ,2)/RR
06192      URT(IJ,3)=URT(IJ,3)/RR
06194      RR=SQRT(URZ(IJ,1)**2+URZ(IJ,2)**2+URZ(IJ,3)**2)
06196      URZ(IJ,1)=URZ(IJ,1)/RR
06198      URZ(IJ,2)=URZ(IJ,2)/RR
06200      URZ(IJ,3)=URZ(IJ,3)/RR
06206 95 CONTINUE
06290      RETURN
06300 980 WRITE(6,981)
06310 981 FORMAT(//,1) *** ERROR IN WIRE INPUT ***
06320      STOP
06330 990 WRITE(6,991)
06340 991 FORMAT(//,1) *** ERROR IN JUNCTION INPUT ***
06350      STOP
06390      END
06580C -----
06590      SUBROUTINE SBOTIN
06600C

```

```

C6610C SKIP BOT INPUTS.
C6620C READ(5,49) NMODE,NPT,NBAND
C6630C READ(5,49) NP
C6640C 49 FORMAT(3I3)
C6750C READ(5,53) (YB,I=1,NP)
C6760C READ(5,53) (XB,I=1,NP)
C6770C 53 FORMAT(10F8.4)
C6870C READ(5,53) BL
C7600C RETURN
C7610C END
C7620C -----
C7430C SUBROUTINE SCAPIN
C7640C
C7650C SKIP CAP INPUTS.
C7660C
C7670C READ(5,1) NC,NPR,NE
C7680C 1 FORMAT(3I3)
C7690C IF(NC.EQ.0) RETURN
C7700C IF(NE.NE.0) NE=NC
C7710C READ(5,2) XC,YC
C7720C 2 FORMAT(10F8.4)
C7730C READ(5,2) (ZC,I=1,NC)
C7740C READ(5,2) (RHOC,I=1,NPR)
C7750C IF(NE.EQ.0) RETURN
C7760C READ(5,2) (ZE,I=1,NE)
C7770C RETURN
C7780C END
C7785C -----
C7790C SUBROUTINE ZLIST(NI,NJ,LI,LJ,NAMI,NAMJ,Z)
C7800C WRITE Z TO THE LINE-PRINTER BY SUBMATRIX BY COLUMNS.
C7810C COMPLEX Z(I)
C7820C DIMENSION LI(1),LJ(1),NAMI(1),NAMJ(1)
C7830C NCOL=C
C7840C DO 100 I=1,NI
C7850C 100 NCOL=NCOL+LI(I)
C7860C NR=1
C7870C DO 300 J=1,NJ
C7880C IF(LJ(J).EQ.0) GO TO 300
C7890C NC=1
C7900C DO 200 I=1,NI
C7910C IF(LI(I).EQ.0) GO TO 200
C7920C WRITE(6,1) NAMI(I),NAMJ(J)
C7930C 1 FORMAT(/,2H Z,1X,A4,3H - ,A4)
C7940C K1=LI(I)
C7950C KJ=LJ(J)
C7960C K1=(NR-1)*NCOL+NC
C7970C DO 150 KK=1,KJ
C7980C K2=K1+K1-1
C7990C WRITE(6,2) (Z(K),K=K1,K2)
C8000C 2 FORMAT(1X,10G11.4)
C8010C 150 K1=K1+NCOL
C8020C NC=NC+LI(I)

```

08030 200 CONTINUE
08040 NR=NR+LJ(J)
08050 300 CONTINUE
08060 RETURN
08070 END
08080/END


```

C01C0BOTZ,P30,T30,CM12CC00,STCX3.
C011CACCOUNT,M03265,BOT2A.
C0120BANNERS(OUTPUT)*J,PUTNAM*DEPT 220*RLD 110-4**
C0130DEFINE(OUTFIL=ZCW)
C0138FTN(R=C,GPT=2)
C0140LDSET(PRESET=INDEF)
C0150LGO.
C0210EXIT.
C0230/END
C0240      PROGRAM BOTZCW(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
C0250      1 OUTFIL,TAPE7=OUTFIL)
C0252C
C0254C BOTZCW GENERATES THE CAP-WIRE/JUNCTION MATRIX.
C0256C
C0258C UNIT 5 IS THE CARD READER.
C0260C UNIT 6 IS THE LINE PRINTER.
C0262C UNIT 7 IS A DISK FILE FOR OUTPUT OF THE Z MATRIX.
C0264C
C0280      COMPLEX U,A3,Z(640C)
C0282      COMMON /WAVE/ RK
C0284      COMMON /BOT1/ NM,MODE,NPT,NBAND
C0286      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
C0288      COMMON /BOT3/ DH(82),SV(82),CV(82)
C0290      COMMON /BOT5/ T(16C),TP(150),TZ(160)
C0292      COMMON /BOT6/ IEDGE,IUNIF
C0294      COMMON /CAP1/ NC,XC,YC,ZC(2)
C0296      COMMON /CAP2/ NPR,RHOC(21),DRHOC(20)
C0298      COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
C0300      COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
C0302      COMMON /EDG1/ NE,7E(2),Z8E(10)
C0304      COMMON /EDG2/ TCE(10),TPCE(10),T8E(10),TP8E(10)
C0306      COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
C0308      1 XW1(100),YW1(100),ZW1(100)
C0310      COMMON /WIRE2/ DHW(100),DXW(100),DYW(100),DZW(100)
C0312      COMMON /WIRE3/ NW,INDW(6),RADW(5)
C0314      COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
C0316      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
C0318      COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
C0320      COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
C0322      COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
C0324      COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
C0326      DIMENSION LI(3),LJ(2),NAMT(3),NAMJ(2)
C0328      DATA NAMT,NAMJ /4HCTT,4HCR,4HEDGE,4HWIRE,4HJUNC/
C0330      U=(0.,1.)
C0332      PI=3.14159265
C0334      ETA=376.707
C0336      READ(5,1) RK
C0338      FORMAT(E15.7)
C0340      WRITE(6,2) RK
C0342      FORMAT(9H1      BK,/,E15.7)
C0344C
C0346C CALL BOTIN

```

```

00460C      CALL CAPIN
00800C
00835C      CALL WIREIN
00836C
00837C      WRITE(7) NC,NPR,NE,NW,NPW,NJ
00838C      NM=(NP-3)/2
00840C      NM2=NM*2
00842C      LR=(NPR-3)/2
00844C      LC=NC+NM*LR
00846C      LC2=LC*2
00848C      LE=NE+NM
00850C      BKL=BK*BKL
00852C      BK2=BK*BK
00854C      LCMJ=(LC2+LE)*(LW+NJ)
00856C      CONSTANTS USED IN SUMMATION OF IMPEOANCE ELEMENTS.
00858C      C1=BK*ETA
00860C      C2=C1
00862C      *** END INITIALIZATION OF PARAMETERS ***
00864C      COMPUTATION OF IMPEOANCE ELEMENTS.
00866C      CAP-WIRE LOOP.
00868C      DO 30 J=1,LW
00870C      L1=(J-1)*LC2+LE1
00872C      J3=(J-1)*4
00874C      J1=INTW(J)-1
00876C      DO 31 IC=1,NC
00878C      DO 31 IP=1,NM
00880C      DO 31 IR=1,LR
00882C      COMPUTE SUBMATRIX INDICES.
00884C      L1=L1+1
00886C      L2=L1+LC
00888C      Z(L1)=0.
00890C      Z(L2)=0.
00892C      DO 70 JJ=1,4
00894C      JR=J1+JJ
00896C      JP=J3+JJ
00898C      IP2=2+(IP-1)
00900C      IP7=4+(IP-1)
00902C      DO 71 IIR=1,4
00904C      IR2=IP2+IIR
00906C      IR7=IP7+IIR
00908C      IR2=2*(IR-1)
00910C      IR7=4*(IR-1)
00912C      DO 71 IIR=1,4
00914C      IR2=IR2+IIR
00916C      IR7=IR7+IIR
00918C      A1=AC*(IP2)*ABS(RHOC(IR2+1)**2-RHOC(IR2)**2)
00920C      R1=RHOC1(IR2)*RC1(IP2)
00922C      XX=XW1(J2)-(XC+RHOC1(IR2)*RC1(IP2)+CPC(IP2))
00924C      YY=YW1(J2)-(YC+RHOC1(IR2)*RC1(IP2)+SPC(IP2))
00926C      ZZ=ZW1(J2)-ZC(IC)
00928C      RH02=XX*XX+YY*YY+ZZ*ZZ

```

```

G2224      RR=SQRT(RHO2)
G2226      A3=CEXP(-U*BK*RR)/4./PI/RR
G2229C     CAP-WIRE IMPEDANCE ELEMENTS COMPUTED.
G2229C     Z(L1) = ZT, Z(L2) = ZR.
G2230C     Z(L1)=Z(L1)+AT*((CV(IP2)*DXW(J2)+SV(IP2)*DYW(J2))*
G2230C     1 T(IP7)+TW(J7)-DHW(J2)+TP(IP7)+TPW(J7)/RHOC1(IR2)*
G2230C     2 /BK2)+TCT(IR7)*A3
G2231C     Z(L2)=Z(L2)+AT*((CPC(IP2)*DXW(J2)+SPC(IP2)*DYW(J2))*
G2231C     1 TCR(IR7)+TW(J7)-DHW(J2)+TPCR(IR7)/RC1(IP2)+
G2231C     2 TCR(IR7)/RT)+TPW(J7)/BK2)+TZ(IP7)*A3
G2237C     71 CONTINUE
G2237C     70 CONTINUE
G2238C     Z(L1)=U+C1*Z(L1)
G2238C     Z(L2)=U+C2*Z(L2)
G2240C     31 CONTINUE
G2240C     30 CONTINUE
G2241C     IF(NJ.EQ.0) GO TO 200
G2241C     NOTE, CAP-JUNCTION NOT INCLUDED.
G2241C     CAP-JUNCTION LOOP.
G2242C     DO 13C J=1,NJ
G2242C     L1=(LW+J-1)+(LC2+LE)
G2242C     DO 131 IC=1,NC
G2242C     DO 131 IP=1,NH
G2242C     DO 131 IR=1,LR
G2247C     COMPUTE SUBMATRIX INDICES.
G2247C     L1=L1+1
G2247C     L2=L1+LC
G2247C     Z(L1)=0.
G2247C     Z(L2)=0.
G2262C     CAP-JUNCTION(WIRE) IMPEDANCE ELEMENTS COMPUTED.
G2262C     Z(L1) = ZT, Z(L2) = ZR.
G2271C     CAP-JUNCTION(DISK) IMPEDANCE ELEMENTS COMPUTED.
G2271C     Z(L1) = ZT, Z(L2) = ZR.
G2272C     131 CONTINUE
G2272C     13C CONTINUE
G2272C     200 CONTINUE
G2272C     IF(NE.EQ.0) GO TO 900
G2273C     EDGE-WIRE LOOP.
G2273C     DO 33C J=1,LW
G2273C     L2=(J-1)+(LC2+LE)+LC2
G2273C     J3=(J-1)+4
G2273C     J1=INTW(J)-1
G2273C     DO 331 IC=1,NC
G2273C     DO 331 IP=1,NH
G2281C     COMPUTE SUBMATRIX INDICES.
G2281C     L2=L2+1
G2281C     Z(L2)=0.
G2281C     DO 370 JJ=1,4
G2281C     J2=J1+JJ
G2281C     J7=J3+JJ
G2281C     IP2=2+(IP-1)
G2281C     IP7=4+(IP-1)

```

```

C2910 DD 371 IIP=1,4
C2920 IP2=IP2+1
C2930 IP7=IP7+1
C2940 IR2=2+IR
C2950 IR7=2+IR(C-1)
C2960 IZ2=2+IR(C-1)
C2970 IZ7=2+IR(C-1)
C2980 DD 371 IIRZ=1,2
C2990 IR2=IR2+1
C3000 IR7=IR7+1
C3010 IZ2=IZ2+1
C3020 IZ7=IZ7+1
C3030 AI=AC(IP2)*ABS(RHOC(IR2+1)**2-RHOC(IR2)**2)
C3040 RI=RHOC1(IR2)*RC1(IP2)
C3050 XX=XW1(J2)-(XC+RHOC1(IR2)*RC1(IP2)*CPC(IP2))
C3060 YY=YW1(J2)-(YC+RHOC1(IR2)*RC1(IP2)*SPC(IP2))
C3070 ZZ=ZW1(J2)-ZC(IC)
C3080 RHQ2=XX*XX+YY*YY+ZZ*ZZ
C3090 RR=SQR(RHQ2)
C3100 A3=CEXP(-U*BK*RR)/4./PI/RR
C3110C EDGE(CAP)-WIRE IMPEDANCE ELEMENTS COMPUTED.
C3120 Z(L2)=Z(L2)+AI*((CPC(IP2)*DXW(J2)+SPC(IP2)*DYW(J2))+
C3130 1 TCE(IR7)*TW(J7)-OHV(J2)+TPCE(IR7)/RC1(IP2)+
C3140 2 TCE(IR7)/RI)*TPW(J7)/BK2)+TZ(IP7)*A3
C3150 AI=OH(IP2)*ABS(ZF(IC)-ZC(IC))/2.
C3160 XX=XW1(J2)-XB1(IP2)
C3170 YY=YW1(J2)-YB1(IP2)
C3180 ZZ=ZW1(J2)-ZB1(I2)
C3190 RHQ2=XX*XX+YY*YY+ZZ*ZZ
C3200 RR=SQR(RHQ2)
C3210C A3=CEXP(-U*BK*RR)/4./PI/RR
C3220C EDGE(BOT)-WIRE IMPEDANCE ELEMENTS COMPUTED.
C3230 Z(L2)=Z(L2)+AI*(D7W(J2)+TBE(I27)*TW(J7)-
C3240 1 OHV(J2)+TPBE(I27)*TPW(J7)/BK2)+TZ(IP7)*A3
C3250 371 CONTINUE
C3260 370 CONTINUE
C3270 Z(L2)=U*C2+Z(L2)
C3280 331 CONTINUE
C3290 330 CONTINUE
C3300 IF(INJ.EQ.0) GO TO 400
C3310C NOTE, EDGE-JUNCTION NOT INCLUDED.
C3320C EDGE-JUNCTION LOOP.
C3330 DO 430 J=1,NJ
C3340 L2=(LW+J-1)*(LC2+LE)+LC2
C3350 DO 431 IC=1,NC
C3360 DO 431 IP=1,NM
C3370C COMPUTE SUBMATRIX INDICES.
C3380 L2=L2+1
C3390 Z(L2)=0.
C3400C EDGE-JUNCTION(WIRE) IMPEDANCE ELEMENTS COMPUTED.
C3410C EDGE-JUNCTION(DISK) IMPEDANCE ELEMENTS COMPUTED.
C3420 431 CONTINUE

```

```

03390 430 CONTINUE
03400C *** END CALCULATION OF Z SURMATRIX ***
03410 900 CONTINUE
03420C PRINT OUT IMPEDANCE MATRIX Z BY SUBMATRIX BY COLUMNS.
03422 LI(1)=LC
03424 LI(2)=LC
03426 LI(3)=LE
03428 LJ(1)=LW
03430 LJ(2)=NJ
03432 CALL ZLIST(3,2,LI,LJ,NAMI,NAMJ,Z)
03440C WRITE Z(M) TO DISK.
03450 WRITE(7) (Z(I),I=1,LCWJ)
03460 998 CONTINUE
03470 STOP
03480 END
-----
03490C SUBROUTINE BOTIN
03500C READ BOT COORDINATES AND COMPUTE BOT SEGMENT ARRAYS.
03510C
03515 COMMON /BOT1/ NMODE,NPT,NBAND
03520 COMMON /BOT2/ NP,RL,YB(83),XB(83),YB1(82),XB1(82)
03525 COMMON /BOT3/ OH(82),SV(82),CV(82)
03530 COMMON /BOT5/ T(160),TP(160),TZ(160)
03535 COMMON /BOT6/ IEDGE,IUNIF
03540 READ(5,49) NMODE,NPT,NBAND
03545 READ(5,49) NP
03550 49 FORMAT(3I3)
03555 WRITE(6,48) NMODE,NPT,NBAND,NP
03560 48 FORMAT(32H NMODE NPT NBAND NP,/,4I8)
03565 READ(5,53) (YB(I),I=1,NP)
03570 READ(5,53) (XB(I),I=1,NP)
03575 53 FORMAT(1GF8.4)
03580 WRITE(6,55)
03585 55 FORMAT(17,3H YB)
03590 WRITE(6,46) (YB(I),I=1,NP)
03595 46 FORMAT(1X,1CF8.4)
03600 WRITE(6,56)
03605 56 FORMAT(17,3H XB)
03610 WRITE(6,46) (XB(I),I=1,NP)
03615C PLOT THE BODY COORDINATES.
03620 CALL PLCTB(XB,YB,NP,41)
03625 READ(5,53) RL
03630 WRITE(6,47) BL
03635 47 FORMAT(17,21H HALF-LENGTH OF BOT =,F12.4)
03640C CHECK IF BOT CONTAINS AN ODD NUMBER OF POINTS.
03645 IF(MOD(NP,2).NE.1) GO TO 980
03650C CHECK IF GENERATING CURVE IS OPEN OR CLOSED.
03655 IEDGE=1
03660 IF((YB(1)-YB(NP)).NE.0..OR.(XB(1)-XB(NP)).NE.0.) GO TO 10
03665 IEDGE=C
03670 YB(NP+1)=YB(2)

```

```

C3675      XB(NP+1)=XB(2)
C3680      YB(NP+2)=YB(3)
C3685      XB(NP+2)=XB(3)
C3690      NP=NP+2
C3695      WRITE(6,66) NP
C3700 66    FORMAT(//,39H BOT GENERATING CURVE IS CLOSED.  NP = ,I3)
C3705 10    CONTINUE
C3710C COMPUTATION OF BODY SEGMENT PARAMETERS.
C3715      DO 57 I=2,NP
C3720      I2=I-1
C3725      RR1=YB(I)-YB(I2)
C3730      RR2=XB(I)-XB(I2)
C3735      DH(I2)=SQRT(RR1*RR1+RR2*RR2)
C3740      XB1(I2)=.5*(XB(I)+XB(I2))
C3745      YB1(I2)=.5*(YB(I)+YB(I2))
C3750      SV(I2)=RR1/DH(I2)
C3755      CV(I2)=RR2/DH(I2)
C3760 57    CONTINUE
C3765C CHECK IF BOT SEGMENTATION IS UNIFORM.
C3770      IUNIF=0
C3775      NP1=NP-1
C3780      DO 60 I=2,NP1
C3785      RR1=DH(I)/DH(1)
C3790      IF(RR1.LT.0.99 .OR. RR1.GT.1.01) GO TO 20
C3795 60    CONTINUE
C3800      IUNIF=1
C3805      WRITE(6,67)
C3810 67    FORMAT(//, ' BOT GENERATING CURVE HAS UNIFORM SEGMENTATION')
C3815 20    CONTINUE
C3820C COMPUTATION OF TRIANGLE FUNCTIONS T.
C3825      NM=(NP-3)/2
C3830      DO 74 J=1,NM
C3835      J2=2*(J-1)+1
C3840      J3=J2+1
C3845      J4=J3+1
C3850      J5=J4+1
C3855      J6=4*(J-1)+1
C3860      J7=J6+1
C3865      J8=J7+1
C3870      J9=J8+1
C3875      DEL1=DH(J2)+DH(J3)
C3880      DEL2=DH(J4)+DH(J5)
C3885      TP(J6)=1./DEL1
C3890      TP(J7)=1./DEL1
C3895      TP(J8)=1./DEL2
C3900      TP(J9)=1./DEL2
C3905      T(J6)=DH(J2)/2./DEL1
C3910      T(J7)=(DH(J2)+DH(J3))/2./DEL1
C3915      T(J8)=(DH(J4)+DH(J5))/2./DEL2
C3920      T(J9)=DH(J5)/2./DEL2
C3925 74    CONTINUE
C3930      NM4=NM+4

```

```

C3935 DO 75 J=1,NM4
C3940 75 TZ(J)=T(J)
C3945 IF(IFDGE.EQ.0) GO TO 76
C3950 TZ(1)=2.-T(1)
C3955 TZ(2)=2.-T(2)
C3960 TZ(NM4-1)=2.-T(NM4-1)
C3965 TZ(NM4)=2.-T(NM4)
C3970 76 CONTINUE
C3975 RETURN
C3980 980 WRITE(6,981)
C3985 981 FORMAT(//,' **** ERROR IN ROT INPUT')
C3990 STOP
C3995 ENC
C4000C *****
C4005 SUBROUTINE PLOTB(X,Y,N,NR)
C4010C *****
C4015C
C4020C WRITTEN 2/14/74 BY J. M. PUTNAM DEPT 220 X23877
C4025C
C4030C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
C4035C N IS THE NUMBER OF POINTS TO BE PLOTTED.
C4040C NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
C4045C NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
C4050C NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
C4055C
C4060 REAL X(1),Y(1),HEAD(10)
C4065 INTEGER LINE(101),BLANK,STAR,PLUS
C4070 DATA BLANK,STAR,PLUS /1H,1H*,1H+/
C4075 NC=51
C4080 N10=(NC-1)/10
C4085 WRITE(6,500)
C4090 500 FORMAT(//,44H BODY COORDINATES, + INDICATES TRIANGLE PEAK)
C4095 WRITE(6,504)
C4100 XMIN=X(1)
C4105 XMAX=X(1)
C4110 YMIN=Y(1)
C4115 YMAX=Y(1)
C4120 DO 6 I=1,N
C4125 IF(X(I).LT.XMIN) XMIN=X(I)
C4130 IF(X(I).GT.XMAX) XMAX=X(I)
C4135 IF(Y(I).LT.YMIN) YMIN=Y(I)
C4140 IF(Y(I).GT.YMAX) YMAX=Y(I)
C4145 6 CONTINUE
C4150 DEL=XMAX-XMIN
C4155 IF(YMAX-YMIN.GT.DEL) DEL=YMAX-YMIN
C4160 XMAX=XMIN+DEL
C4165 YMAX=YMIN+DEL
C4170 DO 5 I=1,N10
C4175 Z=I
C4180 5 HEAD(I)=(XMAX-XMIN)*Z/N10+XMIN
C4185 DY=(YMAX-YMIN)/(NR-1)
C4190 IEDGE=1

```

```

04193 IF(X(I).EQ.X(N) .AND. Y(I).EQ.Y(N)) IEDGE=0
04200 Z=YMAX-DY
04205 YL=Z-DY/2.
04210 DO 7 J=1,NR
04215 DC 8 K=1,NC
04220 8 LINE(K)=BLANK
04225 Z=Z-CY
04230 YU=YL
04235 YL=Z-DY/2.
04240 DO 9 I=1,N
04245 IF(Y(I).GE.YU) GO TO 9
04250 IF(Y(I).LT.YL) GO TO 9
04255 R=(X(I)-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
04260 IF(K.GT.NC) K=NC
04265 LINE(K)=STAR
04270 IF(MOD(I,2).EQ.1) LINE(K)=PLUS
04275 IF(IEGE.EQ.0) GO TO 9
04280 IF(I.EQ.1 .OR. I.EQ.N) LINE(K)=STAR
04285 9 CONTINUE
04290 WRITE(6,508) Z,(LINE(K),K=1,NC)
04295 7 CONTINUE
04300 WRITE(6,504)
04305 WRITE(6,5002)
04310 WRITE(6,507) XMIN,(HEAD(I),I=1,N10)
04315 RETURN
04320 504 FORMAT (1X,14(1H-),1H.,10(5H----.),1H-)
04325 507 FORMAT(10X,11(F10.4))
04330 508 FORMAT(1X,F12.4,1X,1HI,51A1,1HI)
04335 3002 FORMAT(4X,7HYH / KH,4X,1HI,5(9X,1HI))
04340 END
04345C -----
04350C SUBROUTINE CAPIN
04355C
04360C READ CAP INPUTS AND COMPUTE CAP ARRAYS.
04365C
04370 COMMON /BDT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
04375 COMMON /CAP1/ NC,XC,YC,ZC(2)
04380 COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
04385 COMMON /CAP3/ TCR(36),TC1(36),TPCR(36),TPCT(36)
04390 COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
04395 COMMON /EDG1/ NE,ZE(2),ZBE(10)
04400 COMMON /EDG2/ TCE(10),TPCE(10),TBE(10),TPBE(10)
04405 READ(5,1) NC,NPR,NE
04410 1 FORMAT(3I3)
04415 IF(NE.NE.0) NE=NC
04420 WRITE(6,3) NC,NPR,NE
04425 3 FORMAT(10H1 NC,NPR NE,/,3I5,/)
04430 IF(NC.EQ.0) RETURN
04435 READ(5,2) XC,YC
04440 2 FORMAT(10F8.4)
04445 READ(5,2) (ZC(I),I=1,NC)
04450 READ(5,2) (RHOC(I),I=1,NPR)

```

```

04455 IF(NE.NE.0) READ(5,2) (ZE(I),I=1,NE)
04460 WRITE(6,4)
04465 4 FORMAT(37H CAP XC YC ZC ZE,/)
04470 DO 100 I=1,NC
04475 IF(NE.EQ.0) WRITE(6,5) I,XC,YC,ZC(I)
04480 IF(NE.NE.0) WRITE(6,5) I,XC,YC,ZC(I),ZE(I)
04485 100 CONTINUE
04490 5 FORMAT(I4,4X,4F8.4)
04495 WRITE(6,6)
04500 6 FORMAT(//,5H RHOC)
04505 WRITE(6,7) (RHOC(I),I=1,NPR)
04510 7 FORMAT(1X,10F8.4)
04515 IF(MOD(NPR,2).NE.1) GO TO 980
04520 DO 120 I=2,NPR
04525 120 IF(RHOC(I).LE.RHOC(I-1)) GO TO 980
04530C COMPUTATION OF CAP SECTOR PARAMETERS.
04535 DO 47 I=1,NP
04540 RC(I)=SQRT((YB(I)-YC)**2+(XB(I)-XC)**2)
04545 47 CONTINUE
04550 DO 57 I=2,NP
04555 I2=I-1
04560 RR1=YB(I2)-YC
04565 RR2=XB(I2)-XC
04570 RC1(I2)=SQRT(RR1*RR1+RR2*RR2)
04575 AC(I2)=ABS((XB(I)-XC)*(YB(I)+YC)+(XB(I2)-XB(I))*
04580 1 (YB(I2)+YB(I))+(XC-XB(I2))*(YC+YB(I2)))/2.
04585 SPC(I2)=RR1/RC1(I2)
04590 CPC(I2)=RR2/RC1(I2)
04595 57 CONTINUE
04600 DO 67 I=2,NPR
04605 I2=I-1
04610 RHOC1(I2)=(RHOC(I)+RHOC(I2))/2.
04615 DRHOC(I2)=RHOC(I)-RHOC(I2)
04620 67 CONTINUE
04625C COMPUTATION OF CAP TRIANGLE FUNCTIONS.
04630 LC=(NPR-3)/2
04635 DO 74 J=1,LC
04640 J2=2*(J-1)+1
04645 J3=J2+1
04650 J4=J3+1
04655 J5=J4+1
04660 J6=4*(J-1)+1
04665 J7=J6+1
04670 J8=J7+1
04675 J9=J8+1
04680 DEL1=DRHOC(J2)+DRHOC(J3)
04685 DEL2=DRHOC(J4)+DRHOC(J5)
04690 TPCR(J6)=1./DEL1
04695 TPCR(J7)=1./DEL1
04700 TPCR(J8)=1./DEL2
04705 TPCR(J9)=1./DEL2
04710 TCR(J6)=DRHOC(J2)/2./DEL1

```

```

04715   TCR(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
04720   TCR(J8)=(DRHOC(J4)/2.+DRHOC(J5))/DEL2
04725   TCR(J9)=DRHOC(J5)/2./DEL2
04730   74   CONTINUE
04735   LC4=LC+4
04740   DO 75 I=1,LC4
04745   TCT(I)=TCR(I)
04750   TPCT(I)=TPCR(I)
04755   75   CONTINUE
04760   TCT(LC4-1)=2.-TCT(LC4-1)
04765   TCT(LC4)=2.-TCT(LC4)
04770   TPCT(LC4-1)=-TPCT(LC4-1)
04775   TPCT(LC4)=-TPCT(LC4)
04780   IF(RHOC(1).EQ.0.0) GO TO 76
04785   TCT(1)=2.-TCT(1)
04790   TCT(2)=2.-TCT(2)
04795   TPCT(1)=-TPCT(1)
04800   TPCT(2)=-TPCT(2)
04805   76   CONTINUE
04810   IF(INE.EQ.0) RETURN
04815   C COMPUTATION OF EDGE HALF TRIANGLE FUNCTIONS.
04820   DO 80 IC=1,NC
04825   J2=NPR-2
04830   J3=J2+1
04835   J6=2+(IC-1)+1
04840   J7=J6+1
04845   DEL1=DRHOC(J2)+DRHOC(J3)
04850   TPCE(J6)=1./DEL1
04855   TPCE(J7)=1./DEL1
04860   TCE(J6)=DRHOC(J2)/2./DEL1
04865   TCE(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
04870   DEL2=ZE(IC)-ZC(IC)
04875   ZBE(J6)=ZC(IC)+0.25*DEL2
04880   ZBE(J7)=ZC(IC)+0.75*DEL2
04885   IF(DEL2.LT.0.0) GO TO 78
04890   C EDGE IS AT Z=L.
04895   TPBE(J6)=-1./DEL2
04900   TPBE(J7)=-1./DEL2
04905   TBE(J6)=-0.75
04910   TBE(J7)=-0.25
04915   GO TO 80
04920   78   CONTINUE
04925   C EDGE IS AT Z=L.
04930   TPBE(J6)=1./DEL2
04935   TPBE(J7)=1./DEL2
04940   TBE(J6)=-0.75
04945   TBE(J7)=-0.25
04950   80   CONTINUE
04955   RETURN
04960   980  WRITE(6,981)
04965   981  FORMAT(/,' *** ERROR IN CAP INPUT')
04970   STOP

```

```

C4975      END
C4980C-----
C4985      SUBROUTINE WIREIN
C4990C
C4995C      READ WIRE COORDINATES AND COMPUTE WIRE SEGMENT ARRAYS.
C5000C
C5005      COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
C5010      1 XW1(100),YW1(100),ZW1(100)
C5015      COMMON /WIRE2/ NHW(100),DXW(100),DYW(100),DZW(100)
C5020      COMMON /WIRE3/ NW,INDW(6),RADW(5)
C5025      COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
C5030      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
C5035      COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
C5040      COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
C5045      COMMON /JUNC4/ UXJ(10),UYJ(10),UZJ(10)
C5050      COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
C5055      COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
C5060      NW=0
C5065      NJ=C
C5070      LW=0
C5075      READ(5,51) NW,NPW,NJ
C5080C 51  FORMAT(3I3)
C5085      WRITE(6,1) NW,NPW,NJ
C5090C 1  FORMAT(24H1 NW NPW NJ,/,3I8)
C5095      IF(NW.EQ.0) RETURN
C5100      READ(5,53) (XW(I),I=1,NPW)
C5105      READ(5,53) (YW(I),I=1,NPW)
C5110      READ(5,53) (ZW(I),I=1,NPW)
C5115C 53  FORMAT(10F8.4)
C5120      READ(5,52) (INDW(I),I=1,NW)
C5125C 52  FORMAT(10I8)
C5130      INDW(NW+1)=NPW+1
C5135      READ(5,53) (RADW(I),I=1,NW)
C5140      IF(NJ.EQ.0) GO TO 5C
C5145C NOTE, INDJW MUST BE MONOTONIC INCREASING.
C5150      READ(5,52) (INDJW(I),I=1,NJ)
C5155      READ(5,53) (RADD(I),I=1,NJ)
C5160      READ(5,53) (UXJ(I),I=1,NJ)
C5165      READ(5,53) (UYJ(I),I=1,NJ)
C5170      READ(5,53) (UZJ(I),I=1,NJ)
C5175C 5C  IERW=C
C5180      IERJ=C
C5185      WRITE(6,61)
C5190C 61  FORMAT(/,22X,'WIRE COORDINATES',20X,'JUNCTION PARAMETERS',/,
C5195      1 26X,'IW',4X,'XW',4X,'YW',6X,'ZW',
C5200      2 10X,'IJ',4X,'RADD',5X,'UXJ',5X,'UYJ',5X,'UZJ')
C5205      1J=1
C5210C THIS LOOP LISTS WIRE/JUNCTION POINTS, WHILE CHECKING THE FOLLOWING:
C5215C 1) EACH WIRE MUST CONTAIN AN ODD NUMBER OF POINTS.
C5220C 2) EACH JUNCTION MUST EITHER START OR TERMINATE A WIRE.
C5225C 3) CHECK THAT ALL JUNCTION POINTS ARE FOUND.
C5230C DO 1CC IW=1,NW

```

AD-A118 498

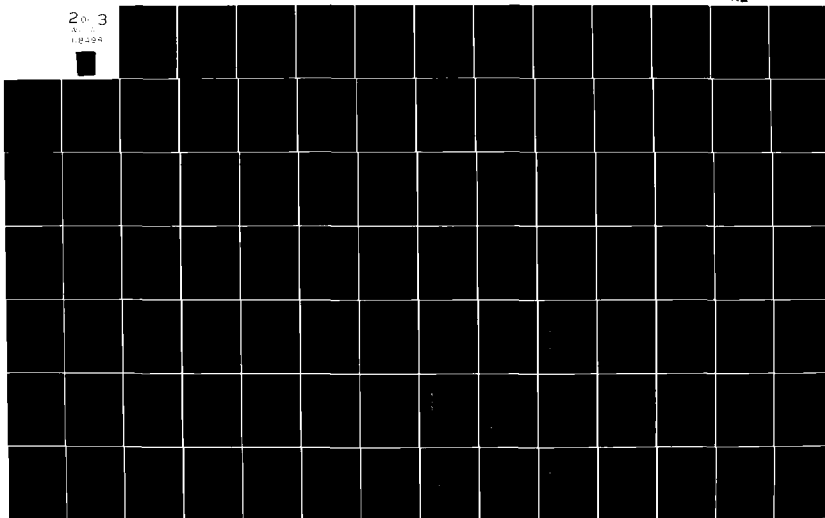
MCDONNELL DOUGLAS RESEARCH LABS ST LOUIS MO
ALGORITHM FOR SURFACE OF TRANSLATION ATTACHED RADIATORS (A-STAR--ETC (U))
MAY 82 L N MEDGYESI-MITSCHANG, J M PUTNAM F30602-80-C-0106
RADC-TR-82-113-VOL-3 NL

UNCLASSIFIED

20-3

AD-A118 498

1-2484



```

05235      WRITE(6,62) IW,RADW(IW)
05240 62    FORMAT(2X,'WIRE',I3,' RADW=',F8.4)
05245      I1=INDW(IW)
05250      I2=INDW(IW+1)-1
05255C CHECK FOR AN ODD NUMBER OF POINTS ON WIRE IW.
05260      IF(MOD(I2-I1+1,2).NE.1) IERW=1
05265      DO 9C I=I1,I2
05270      WRITE(6,63) I,XW(I),YW(I),ZW(I)
05275 63    FORMAT(25X,I3,3F8.4)
05280C CHECK IF WIRE POINT I IS A JUNCTION POINT.
05285      IF(IJ.GT.NJ) GO TO 90
05290      IF(INDJW(IJ).NE.I) GO TO 90
05295C CHECK THAT JUNCTION POINT IJ IS AT THE START OR END OF WIRE.
05300      IF(I.NE.I1.AND. I.NE.I2) IERJ=1
05305      WRITE(6,64) IJ,RADD(IJ),UXJ(IJ),UYJ(IJ),UZJ(IJ)
05310 64    FORMAT(' ',58X,I3,4F8.4)
05315C DETERMINE DIRECTION IN WHICH WIRE LEAVES JUNCTION POINT.
05320      INDTJ(IJ)=1
05325      IF(I.EQ.I2) INDTJ(IJ)=-1
05330C COMPUTE JUNCTION PARAMETERS.
05335      RADJ(IJ)=RADW(IW)
05340      XJ(IJ)=XW(I)
05345      YJ(IJ)=YW(I)
05350      ZJ(IJ)=ZW(I)
05355      IJ=IJ+1
05360 90    CONTINUE
05365 100   CONTINUE
05370C CHECK FOR WIRE OR JUNCTION INPUT ERRORS.
05375      IF(IERW.NE.C) GO TO 980
05380      IF(IJ-1.NE.NJ .OR. IERJ.NE.O) GO TO 990
05385C COMPUTATION OF WIRE SEGMENT PARAMETERS.
05390      DO 57 I=2,NPW
05395      I2=I-1
05400      DXW(I2)=XW(I)-XW(I2)
05405      DYW(I2)=YW(I)-YW(I2)
05410      DZW(I2)=ZW(I)-ZW(I2)
05415      DHW(I2)=SQRT(DXW(I2)**2+DYW(I2)**2+DZW(I2)**2)
05420      XW1(I2)=0.5*(XW(I)+XW(I2))
05425      YW1(I2)=0.5*(YW(I)+YW(I2))
05430      ZW1(I2)=0.5*(ZW(I)+ZW(I2))
05435      57 CONTINUE
05440C COMPUTATION OF WIRE TRIANGLE FUNCTIONS TW.
05445      LW=0
05450      DO 75 IW=1,NW
05455      I1=INDW(IW)
05460      I2=INDW(IW+1)-1
05465      LW1=(I2-I1-2)/2
05470      DO 74 J=1,LW1
05475      LW=LW+1
05480      J2=2*(J-1)+I1
05485      J3=J2+1
05490      J4=J3+1

```

```

C5495      J5=J4+1
C5500      J6=4*(LW-1)+1
C5505      J7=J6+1
C5510      J8=J7+1
C5515      J9=J8+1
C5520      INDJW(LW)=J2
C5525      DEL1=DHW(J2)+DHW(J3)
C5530      DEL2=DHW(J4)+DHW(J5)
C5535      TPW(J6)=1./DEL1
C5540      TPW(J7)=1./DEL1
C5545      TPW(J8)=-1./DEL2
C5550      TPW(J9)=-1./DEL2
C5555      TW(J6)=DHW(J2)/2./DEL1
C5560      TW(J7)=(DHW(J2)+DHW(J3)/2.)/DEL1
C5565      TW(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
C5570      TW(J9)=DHW(J5)/2./DEL2
C5575      74 CONTINUE
C5580      75 CONTINUE
C5585C     NOTE, LW=(NPW-NW)/2-NW
C5590      IF(NJ.EQ.0) RETURN
C5595C     COMPUTATION OF JUNCTION HALF TRIANGLE FUNCTIONS TJ.
C5600      DO 85 IJ=1,NJ
C5605      IF(INDJ(IJ).GT.0) GO TO 80
C5610C     JUNCTION IS AT THE END OF A WIRE.
C5615      J2=INDJW(IJ)-2
C5620      J3=J2+1
C5625      J6=2*(IJ-1)+1
C5630      J7=J6+1
C5635      INDJ(IJ)=J2
C5640      DEL1=-TW(J2)-DHW(J3)
C5645      TPJ(J6)=1./DEL1
C5650      TPJ(J7)=1./DEL1
C5655      TJ(J6)=DHW(J2)/2./DEL1
C5660      TJ(J7)=(DHW(J2)+DHW(J3)/2.)/DEL1
C5665      GO TO 85
C5670C     JUNCTION IS AT THE START OF A WIRE.
C5675      80 J4=INDJW(IJ)
C5680      J5=J4+1
C5685      J8=2*(IJ-1)+1
C5690      J9=J8+1
C5695      INDJ(IJ)=J4
C5700      DEL2=DHW(J4)+DHW(J5)
C5705      TPJ(J8)=-1./DEL2
C5710      TPJ(J9)=-1./DEL2
C5715      TJ(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
C5720      TJ(J9)=DHW(J5)/2./DEL2
C5725      85 CONTINUE
C5730C     COMPUTE UNIT VECTORS FOR JUNCTION(DISK).
C5735      DO 95 IJ=1,NJ
C5740C     NORMAL UNIT VECTOR (UJ).
C5745      RR=SQRT(UXJ(IJ)**2+UYJ(IJ)**2+UZJ(IJ)**2)
C5750      UXJ(IJ)=UXJ(IJ)/RR

```

```

05755      UYJ(IJ)=UYJ(IJ)/RR
05760      UZJ(IJ)=UZJ(IJ)/RR
05765C FIND 2 ORTHOGONAL UNIT VECTORS IN THE PLANE OF THE DISK (UJ1 & UJ2).
05770      UXJ1(IJ)=0.0
05775      UYJ1(IJ)=0.0
05780      UZJ1(IJ)=0.0
05785      UXJ2(IJ)=0.0
05790      UYJ2(IJ)=0.0
05795      UZJ2(IJ)=0.0
05800      IF(UXJ(IJ).EQ.0.0) GO TO 91
05805C FIND INTERSECTION WITH X-Y PLANE.
05810      UYJ1(IJ)=1.0
05815      UXJ1(IJ)=-UYJ(IJ)/UXJ(IJ)
05820C FIND INTERSECTION WITH X-Z PLANE.
05825      UZJ2(IJ)=1.0
05830      UXJ2(IJ)=-UZJ(IJ)/UXJ(IJ)
05835      GO TO 94
05840 91 IF(UYJ(IJ).EQ.0.0) GO TO 92
05845C FIND INTERSECTION WITH Y-Z PLANE.
05850      UZJ1(IJ)=1.0
05855      UYJ1(IJ)=-UZJ(IJ)/UYJ(IJ)
05860C FIND INTERSECTION WITH X-Y PLANE.
05865      UXJ2(IJ)=1.0
05870      UYJ2(IJ)=-UXJ(IJ)/UYJ(IJ)
05875      GO TO 94
05880 92 IF(UZJ(IJ).EQ.0.0) GO TO 94
05885C FIND INTERSECTION WITH X-Z PLANE.
05890      UXJ1(IJ)=1.0
05895      UZJ1(IJ)=-UXJ(IJ)/UZJ(IJ)
05900C FIND INTERSECTION WITH Y-Z PLANE.
05905      UYJ2(IJ)=1.0
05910      UZJ2(IJ)=-UYJ(IJ)/UZJ(IJ)
05915 94 CONTINUE
05920      RR=SQRT(UXJ1(IJ)**2+UYJ1(IJ)**2+UZJ1(IJ)**2)
05925      UXJ1(IJ)=UXJ1(IJ)/RR
05930      UYJ1(IJ)=UYJ1(IJ)/RR
05935      UZJ1(IJ)=UZJ1(IJ)/RR
05940      RR=SQRT(UXJ2(IJ)**2+UYJ2(IJ)**2+UZJ2(IJ)**2)
05945      UXJ2(IJ)=UXJ2(IJ)/RR
05950      UYJ2(IJ)=UYJ2(IJ)/RR
05955      UZJ2(IJ)=UZJ2(IJ)/RR
05960 95 CONTINUE
05965      RETURN
05970 980 WRITE(6,981)
05975 981 FORMAT(/, ' *** ERROR IN WIRE INPUT ')
05980      STOP
05985 990 WRITE(6,991)
05990 991 FORMAT(/, ' *** ERROR IN JUNCTION INPUT ')
05995      STOP
06000      END
08550C -----
08555      SUBROUTINE ZLIST(NI,NJ,LI,LJ,NAMI,NAMJ,Z)

```

```

08560C WRITE Z TO THE LINE-PRINTER BY SUBMATRIX BY COLUMNS.
08565      COMPLEX Z(1)
08570      DIMENSION LI(1),LJ(1),NAMI(1),NAMJ(1)
08575      NCOL=0
08580      DO 100 I=1,NI
08585 10C  NCOL=NCOL+LI(I)
08590      NR=1
08595      DO 300 J=1,NJ
08600      IF(LJ(J).EQ.0) GO TO 300
08605      NC=1
08610      DO 200 I=1,NI
08615      IF(LI(I).EQ.0) GO TO 200
08620      WRITE(6,1) NAMI(I),NAMJ(J)
08625 1      FORMAT(//,2H Z,1X,A4,3H - ,A4)
08630      KI=LI(I)
08635      KJ=LJ(J)
08640      K1=(NR-1)*NCOL+NC
08645      DO 150 KK=1,KJ
08650      K2=K1+KI-1
08655      WRITE(6,2) (Z(K),K=K1,K2)
08660 2      FORMAT(1X,10G11.4)
08665 15C  K1=K1+NCOL
08670      NC=NC+LI(I)
08675 200  CONTINUE
08680      NR=NR+LJ(J)
08685 300  CONTINUE
08740      RETURN
08750      END
08760/EOR

```



```

C01C0B0TI,P30,T1C0,CM145000,STCX3.
C0110ACCCUNT,NG326S,B0T2A.
C0120BANNERS(OUTPUT) *J. PUTNAM*DEPT 220*BLD 110-4**
C0130ATTACH(INFIL=ZSSX)
C0135DEFINE(OUTFIL=YSSX)
C014CFTN(R=0,OPT=2)
C0170LDSET(PRESET=INDEF)
C0180LGO.
C0220/EOR
C0230      PROGRAM BOTINV(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
C0231      1 INFIL,TAPE1=INFIL,OUTFIL,TAPE7=OUTFIL)
C0232C
C0233C      BOTINV INVERTS THE BOT Z MATRIX, YIELDING THE BOT Y MATRIX.
C0234C
C0235C      UNIT 5 IS THE CARD READER.
C0236C      UNIT 6 IS THE LINE PRINTER.
C0237C      UNIT 1 IS A DISK FILE CONTAINING THE Z MATRIX.
C0238C      UNIT 7 IS A DISK FILE FOR OUTPUT OF THE Y MATRIX.
C0239C
C0250      COMPLEX Z(12544),ZI(1792),LOAD(16)
C0255      DIMENSION WGHT(16)
C0260      COMPLEX Z1,Z2,Z3,Z4
C0265      COMPLEX Z1T,Z2T,Z3T,Z4T
C0270      DIMENSION NZ(31)
C0280      COMMON NM,JK(4),LR(112)
C0285      NFOUNT=0
C0290      READ(5,1) NMODE,NBAND
C0291      1      FORMAT(2I3)
C0292      READ(11,NP)
C0293      NBAND1=NBAND-1
C0305      WRITE(6,6) NP,NMODE,NBAND
C0310      6      FORMAT(/,32H          NP          NMODE          NBAND,/,3I10)
C0326      LS=NP-3
C0330      NM=LS/2
C0340      LSS=LS*LS
C0342C      READ SURFACE IMPEDANCE LOADING.
C0345      READ(5,8) (LOAD(I),I=1,NM)
C0346      8      FORMAT(10F8.4)
C0347      NM1=NM+1
C0348      READ(5,8) (LOAD(I),I=NM1,LS)
C0349      WRITE(6,9)
C0350      9      FORMAT(/,1 T AND 7 SURFACE IMPEDANCE LOADING (COMPLEX)*)
C0351      WRITE(6,59) (LOAD(I),I=1,NM)
C0352      59      FORMAT(/,1X,10F8.4)
C0353      WRITE(6,59) (LOAD(I),I=NM1,LS)
C0354      READ(1) (WGHT(I),I=1,NM)
C0355      READ(1) (WGHT(I),I=NM1,LS)
C0356      DO 600 I=1,LS
C0357      600      LOAD(I)=LOAD(I)*WGHT(I)
C0362      I3=NM*LS
C0364      I4=I3+NM
C0370      JK(1)=1

```

```

00380      JK(2)=NM+1
00390      JK(3)=2*NM*NM+1
00395      JK(4)=JK(3)+NM
00398      WRITE(7) NMODE,NP,0,0,0,0,0,0
00400      WRITE(7) 0,C
00401      WRITE(6,2)
00402 2      FORMAT(//,46H THE MINIMUM ROTINV ARRAY DIMENSIONS FOR THIS,
00403      1 24H PROBLEM ARE AS FOLLOWS:,//,
00404      2 49H      Z      ZI      LOADWGT      NZ      LR,/)
00405      IF(NBAND.LE.0) STOP
00410      IF(NBAND.EQ.1) GO TO 40C
00420      IF(NBAND.LT.2*NMODE-1) GO TO 200
00430C ***** INVERT THE ENTIRE Z ROT MATRIX *****
00440      LL=LS*(2*NMODE-1)
00450      L3=NM*LL
00460      L4=L3+NM
00461      K1=LL*LL
00462      K2=LSS
00463      K3=LS
00464      K4=0
00465      K5=LL
00466      WRITE(6,3) K1,K2,K3,K4,K5
00467 3      FORMAT(5I10)
00468      WRITE(6,53)
00470 53      FORMAT(//,22H FULL MATRIX INVERSION)
00472      NEXP=NMODE*NMODE
00474 5      READ(1) M,N
00480      IF(EOF(1)) 100,10
00490 10      READ(1) (ZI(I),I=1,LSS)
00500      IF(IABS(M).GE.NMODE .OR. IABS(N).GE.NMODE) GO TO 5
00505      NFOUND=NFOUND+1
00506C ADD SURFACE IMPEDANCE TO DIAGONAL.
00507      I=1
00508      DO 11 J=1,LS
00510      IF(J.GT.NM .AND. M*N.EQ.0) GO TO 11
00511      ZI(I)=ZI(I)+LOAD(J)
00512 11      I=I+LS+1
00515      K1=(N+NMODE-1)*LL*LS+(M+NMODE-1)*LS
00520      K2=(M+NMODE-1)*LL*LS+(N+NMODE-1)*LS
00530      K3=(-N+NMODE-1)*LL*LS+(-M+NMODE-1)*LS
00540      K4=(-M+NMODE-1)*LL*LS+(-N+NMODE-1)*LS
00550C INSERT ZI INTO THE Z MATRIX, USING THE SYMMETRIES.
00560      DO 20 J=1,NM
00570      II=(J-1)*LS
00575      IIT=J-LS
00580      DO 15 I=1,NM
00590      II=II+1
00595      IIT=IIT+LS
00600      K1=K1+1
00610      K2=K2+1
00620      K3=K3+1
00630      K4=K4+1

```

```

00635C ZTT , ZTZ , ZTZ , AND ZZZ MATRIX ELEMENTS.
00640      Z1=Z1(I1)
00650      Z2=Z1(I1+NM)
00660      Z3=Z1(I1+I3)
00670      Z4=Z1(I1+I4)
00671C TRANSPOSE ELEMENTS.
00672      Z1T=Z1(I1T)
00674      Z2T=Z1(I1T+NM)
00676      Z3T=Z1(I1T+I3)
00678      Z4T=Z1(I1T+I4)
00680C Z(M,N).
00690      Z(K1)=Z1
00700      Z(K1+NM)=Z2
00710      Z(K1+L3)=Z3
00720      Z(K1+L4)=Z4
00730      IF(M.EQ.N) GO TO 12
00740C Z(N,M).
00750      Z(K2)=Z1T
00760      Z(K2+NM)=-Z3T
00770      Z(K2+L3)=-Z2T
00780      Z(K2+L4)=Z4T
00790      IF(M.EQ.-N) GO TO 15
00800C Z(-M,-N).
00810      Z(K3)=Z1
00820      Z(K3+NM)=-Z2
00830      Z(K3+L3)=-Z3
00840      Z(K3+L4)=Z4
00850      IF(M.EQ.-N) GO TO 15
00860C Z(-N,-M).
00870      Z(K4)=Z1T
00880      Z(K4+NM)=Z3T
00890      Z(K4+L3)=Z2T
00900      Z(K4+L4)=Z4T
00910      15 CONTINUE
00920      K1=K1+LL-NM
00930      K2=K2+LL-NM
00940      K3=K3+LL-NM
00950      K4=K4+LL-NM
00960      20 CONTINUE
00970      GO TO 5
00980      100 CONTINUE
00985      IF(NFCUND.NE.NEXP) GO TO 490
00987C INVERT THE Z MATRIX.
00990      CALL LINEQ(LL,?,LR)
01000C OUTPUT Z MATRIX BY MODE NUMBERS.
01010      KMODE=2+NMDE-1
01020      DO 15C NN=1,KMODE
01030      N=-NMDE+NN
01040      DO 150 MM=1,KMODE
01050      M=-NMDE+MM
01060      II=0
01070      K=(N+NMDE-1)*LL+LS+(M+NMDE-1)*LS

```

```

0108C      DO 125 J=1,LS
01090      DO 12C I=1,LS
01100      II=II+1
01110      K=K+1
01120      ZI(II)=Z(K)
01130 12C      CONTINUE
01140      K=K+LL-LS
01150 125      CONTINUE
01160      CALL LIST(M,N,ZI)
01170 150      CONTINUE
01180      GO TO 5CC
01190C ***** INVERT THE BANDED Z BOT MATRIX *****
01200 2CC      KMODE=2*NMODE-1
01210      K1=NBAND
01220      K2=KMODE-NBAND+2
01230      KZ=KMODE-K1+1
01240      NZ(1)=0
01250      DO 2C2 I=2,KMODE
01260      I1=I-1
01270      IF(I.LE.K1) KZ=KZ-1
01280      IF(I.GT.K2) KZ=KZ+1
01290 2C2      NZ(I)=NZ(I1)+KZ
01291      K1=LSS*(KMODE*(2*NBAND-1)-NBAND*(NBAND-1))
01292      K2=KMODE*LSS
01293      K3=LS
01294      K4=KMODE
01295      K5=LS
01296      WRITE(6,3) K1,K2,K3,K4,K5
01298      WRITE(6,54)
01300 54      FORMAT(/,24H BANDED MATRIX INVERSION)
01302      NEXP=NMODE*NMODE-MOD(NBAND1,2)*(NMODE-(NBAND1+1)/2)**2
01304      1 +(MOD(NBAND1,2)-1)*(NMODE-NBAND1/2-1)*(NMODE-NBAND1/2)
01306 205      READ(1) M,N
01310      IF(EQF(1)) 300,210
01320 210      IF(IABS(M-N).GE.NBAND) GO TO 300
01330      READ(1) (ZI(I),I=1,LSS)
01340      IF(IABS(M).GE.NMODE .OR. IABS(N).GE.NMODE) GO TO 205
01345      NFOUND=NFOUND+1
01346C ADD      SURFACE IMPEDANCE TO DIAGONAL.
01347      I=1
01348      DO 211 J=1,LS
01350      IF(J.GT.NM .AND. M*N.EQ.0) GO TO 211
01351      ZI(I)=ZI(I)+LOAD(J)
01352 211      I=I+LS+1
01355      K1=((N+NMODE-1)*KMODE+(M+NMODE-1)-NZ(N+NMODE))*LSS
01360      K2=((M+NMODE-1)*KMODE+(N+NMODE-1)-NZ(M+NMODE))*LSS
01370      K3=((-N+NMODE-1)*KMODE+(-M+NMODE-1)-NZ(-N+NMODE))*LSS
01380      K4=((-M+NMODE-1)*KMODE+(-N+NMODE-1)-NZ(-M+NMODE))*LSS
01385C INSERT ZI INTO THE Z MATRIX, USING THE SYMMETRIES.
01390      DO 220 J=1,NM
01400      II=(J-1)*LS
01405      IIT=J-LS

```

```

01410      DO 215 I=1,NM
01420      II=I+1
01425      IIT=IIT+LS
01430      K1=K1+1
01440      K2=K2+1
01450      K3=K3+1
01460      K4=K4+1
01465C ZTT , ZZT , ZTZ , AND ZZZ MATRIX ELEMENTS.
01470      Z1=ZI(II)
01480      Z2=ZI(II+NM)
01490      Z3=ZI(II+I3)
01500      Z4=ZI(II+I4)
01505C TRANSPOSE ELEMENTS.
01510      Z1T=ZI(IIT)
01515      Z2T=ZI(IIT+NM)
01520      Z3T=ZI(IIT+I3)
01525      Z4T=ZI(IIT+I4)
01530C Z(M,N).
01535      Z(K1)=Z1
01540      Z(K1+NM)=Z2
01545      Z(K1+I3)=Z3
01550      Z(K1+I4)=Z4
01555      IF(M.EQ.N) GO TO 212
01560C Z(N,M).
01565      Z(K2)=Z1T
01570      Z(K2+NM)=-Z3T
01575      Z(K2+I3)=-Z2T
01580      Z(K2+I4)=Z4T
01585      IF(M.EQ.-N) GO TO 215
01590C Z(-M,-N).
01595      Z(K3)=Z1
01600      Z(K3+NM)=-Z2
01605      Z(K3+I3)=-Z3
01610      Z(K3+I4)=Z4
01615      IF(M.EQ.-N) GO TO 215
01620C Z(-N,-M).
01625      Z(K4)=Z1T
01630      Z(K4+NM)=Z3T
01635      Z(K4+I3)=Z2T
01640      Z(K4+I4)=Z4T
01645      IF(M.EQ.-N) GO TO 215
01650      CONTINUE
01655      K1=K1+NM
01660      K2=K2+NM
01665      K3=K3+NM
01670      K4=K4+NM
01675      CONTINUE
01680      GO TO 205
01685      CONTINUE
01690      IF(NFOUND.NE.NEXP) GO TO 490
01695C INVERT THE Z MATRIX AND THEN LIST THE INVERTED MATRIX.
01700      CALL INVBAN(LS,NMODE,NRAND,NZ,Z,ZI,LOAD)
01705      GO TO 500

```

```

01840C ***** INVERT THE DIAGONAL Z BOT MATRIX *****
01851 400 K1=0
01852 K2=LSS
01853 K3=LS
01854 K4=0
01855 K5=LS
01856 WRITE(6,3) K1,K2,K3,K4,K5
01858 WRITE(6,55)
01860 55 FORMAT(/,26H DIAGONAL MATRIX INVERSION)
01862 NEXP=NMODE
01864 405 READ(1) M,N
01870 IF(ECF(1)) 500,410
01880 410 IF(IABS(M-N).GT.0) GO TO 500
01890 IF(M.GE.NMODE) GO TO 500
01900 READ(1) (ZI(I),I=1,LSS)
01905 NFOUND=NFOUND+1
01906C ADD SURFACE IMPEDANCE TO DIAGONAL.
01907 I=1
01908 DO 411 J=1,LS
01910 IF(J.GT.NM .AND. M*N.EQ.0) GO TO 411
01911 ZI(I)=ZI(I)+LOAD(J)
01912 411 I=I+LS+1
01913 CALL LINEQ(LS,ZI,LR)
01914 CALL LIST(M,N,ZI)
01915 IF(M.EQ.0) GO TO 405
01922C COMPUTE Z(-M,-N) INVERSE FROM Z(M,N) INVERSE.
01924C NOTE, THE OFF DIAGONAL SUBMATRICES CHANGE SIGNS.
01930 DO 430 J=1,NM
01931 II=(J-1)*LS
01932 DO 425 I=1,NM
01933 II=II+1
01934 ZI(II+NM)=-ZI(II+NM)
01935 ZI(II+I3)=-ZI(II+I3)
01936 425 CONTINUE
01937 430 CONTINUE
01940 CALL LIST(-M,-N,ZI)
01950 GO TO 405
01952 490 WRITE(6,7)
01954 7 FORMAT(/,41H ERROR IN THE NUMBER OF SUBMATRICES FOUND)
01960 500 CONTINUE
01965 WRITE(6,4) NEXP,NFOUND
01966 4 FORMAT(/,33H NUMBER OF SUBMATRICES EXPECTED =,I5,/,
01967 1 30H NUMBER OF SUBMATRICES FOUND =,I8)
01970 STOP
01980 END
02185C *****
02190 SUBROUTINE LINEQ(LL,C,LR)
02195C *****
02197C COMPLEX MATRIX INVERSION ROUTINE.
02200 COMPLEX C(1),STOR,STO,ST,S
02210 DIMENSION LR(1)
02220 DO 2C I=1,LL

```

```

G2230 LR(I)=I
G2240 20 CONTINUE
G2250 M1=0
G2260 DO 10 M=1,LL
G2270 K=M
G2280 K2=M1+K
G2290 S1=CABS(C(K2))
G2300 DO 2 I=M,LL
G2310 K1=M1+I
G2320 S2=CABS(C(K1))
G2330 IF(S2-S1) 2,2,6
G2340 6 K=I
G2350 S1=S2
G2360 2 CONTINUE
G2370 LS=LR(M)
G2380 LR(M)=LR(K)
G2390 LR(K)=LS
G2400 K2=M1+K
G2410 STOR=C(K2)
G2420 J1=C
G2430 DO 7 J=1,LL
G2440 K1=J1+K
G2450 K2=J1+M
G2460 STO=C(K1)
G2470 C(K1)=C(K2)
G2480 C(K2)=STO/STOR
G2490 J1=J1+LL
G2500 7 CONTINUE
G2510 K1=M1+M
G2520 C(K1)=1./STOR
G2530 DO 11 I=1,LL
G2540 IF(I-M) 12,11,12
G2550 12 K1=M1+I
G2560 ST=C(K1)
G2570 C(K1)=0.
G2580 J1=C
G2590 DO 10 J=1,LL
G2600 K1=J1+I
G2610 K2=J1+M
G2620 C(K1)=C(K1)-C(K2)*ST
G2630 J1=J1+LL
G2640 10 CONTINUE
G2650 11 CONTINUE
G2660 M1=M1+LL
G2670 18 CONTINUE
G2680 J1=C
G2690 DO 9 J=1,LL
G2700 IF(J-LR(J)) 14,9,14
G2710 14 LRJ=LR(J)
G2720 J2=(LRJ-1)*LL
G2730 21 DO 13 I=1,LL
G2740 K2=J2+I

```

```

C2750      K1=J1+I
C2760      S=C(R2)
C2770      C(K2)=C(K1)
C2780      C(K1)=S
C2790      13 CONTINUE
C2800      LR(J)=LR(LRJ)
C2810      LR(LRJ)=LRJ
C2820      IF(J-LR(J)) 14,8,14
C2830      8 J1=J1+LL
C2840      9 CONTINUE
C2850      RETURN
C2860      END
C2865C *****
C2870      SUBROUTINE LIST(M,N,ZI)
C2875C *****
C2880C PRINT Z(M,N) ON THE LINE PRINTER AND WRITE TO DISK FILE.
C2890      COMPLEX ZI(1)
C2900      COMMON NM,JK(4)
C2910      LS=2*NM
C2920      LSS=LS*LS
C2930      IF(M.NE.N) GO TO 2C0
C2940      2 WRITE(6,2) M,N
C2950      FORMAT(4H1 M=,I3,4H N=,I3,/)
C2960      DO 14C J=1,4
C2970      K1=JK(J)
C2980      3 WRITE(6,3) J
C2990      FORMAT(2H Y,I1)
C3000      DO 140 I=1,NM
C3010      K2=K1+NM-1
C3020      4 WRITE(6,4) (ZI(K),K=K1,K2)
C3030      FORMAT(1X,10G11.4)
C3040      K1=K1+LS
C3050      140 CONTINUE
C3060      2C0 CONTINUE
C3070      WRITE(7) M,N
C3080      WRITE(7) (ZI(I),I=1,LSS)
C3090      RETURN
C3100      END
C3105C *****
C3110C SUBROUTINE INVRAN(LS,NMODE,NBAND,NZ,A,Z,WORK)
C3115C *****
C3120C THIS ROUTINE IS A MODIFICATION OF A STANDARD INVERSION ROUTINE
C3130C (FOR SMALL MATRICES) USING LU DECOMPOSITION WITHOUT PIVOTING.
C3140C ALL REFERENCES TO ELEMENTS WERE CHANGED TO SUBMATRIX REFERENCES,
C3150C AS DESCRIBED IN THE BOTINV PROGRAM DESCRIPTION. ALL COMMENTS IN
C3160C THIS PROGRAM REFERENCE THE ORIGINAL PROGRAM.
C3170C THIS ROUTINE WILL INVERT A Banded MATRIX. NBAND-1 IS THE NUMBER OF
C3180C BANDS ABOVE AND BELOW THE MAIN DIAGONAL. ONLY THE Banded PORTION
C3190C OF MATRIX A IS STORED BY COLUMNS.
C3200C COMPLEX A(1),Z(1),WORK(1)

```

```

03130      INTEGER NZ(1)
03140C    NZ(I) = NZ(I-1) + THE NUMBER OF ZEROES BELOW THE BAND IN COLUMN I-1
03150C    + THE NUMBER OF ZEROES ABOVE THE BAND IN COLUMN I, WHERE NZ(1)=0.
03155      COMMON NM,JK(4),LR(1)
03160      K1=NBAND
03170      K2=N-NBAND+2
03180C
03190C    NOTE, THE LOCATION OF ELEMENT A(I,J) IN THE BANDED PORTION OF MATRIX
03200C    A IS GIVEN BY N * (J-1) + I - NZ(J).
03210C
03220C    PERFORM THE LU DECOMPOSITION OF A, WHERE U HAS UNIT DIAGONAL ELEMENTS.
03230      N=2*NMODE-1
03240      LSS=LS*LS
03250      DO 100 J=1,N
03260        J1=J-1
03270        JP=J+1
03280        IL=(N*J1+J1-NZ(J))*LSS+1
03290C    CALCULATE L(I,J).
03300        DO 20 I=J,N
03310          IF(IABS(I-J).GE.NBAND) GO TO 20
03320          IF(J.EQ.1) GO TO 19
03330          DO 15 K=1,J1
03340            IF(IABS(I-K).GE.NBAND .OR. IABS(K-J).GE.NBAND) GO TO 15
03350            KL=(N*(K-1)+I-NZ(K)-1)*LSS+1
03360            KU=(N*(J-1)+K-NZ(J)-1)*LSS+1
03370            CALL MULTS(LS,A(IL),A(KL),A(KU))
03380      15 CONTINUE
03390      19 CONTINUE
03400        IF(I.EQ.J) CALL LINEQ(LS,A(IL),LR)
03410      20 IL=IL+LSS
03420C    CALCULATE U(J,I).
03430        IF(J.EQ.N) GO TO 100
03440        IL=(N*(J-1)+J-NZ(J)-1)*LSS+1
03450        DO 30 I=JP,N
03460          IF(IABS(I-J).GE.NBAND) GO TO 30
03470          IU=(N*(I-1)+J-NZ(I)-1)*LSS+1
03480          IF(J.EQ.1) GO TO 29
03490          DO 25 K=1,J1
03500            IF(IABS(I-K).GE.NBAND .OR. IABS(K-J).GE.NBAND) GO TO 25
03510            KL=(N*(K-1)+I-NZ(K)-1)*LSS+1
03520            KU=(N*(J-1)+K-NZ(J)-1)*LSS+1
03530            CALL MULTS(LS,A(IU),A(KL),A(KU))
03540      25 CONTINUE
03550      29 CALL MULT(LS,WORK,A(IL),A(IU))
03560      30 CONTINUE
03570      100 CONTINUE
03580C    FIND THE INVERSE OF A BY COLUMNS. SOLVE LZ=B, WHERE B IS A COLUMN
03590C    OF THE IDENTITY MATRIX. THEN SOLVE UX=Z. X WILL THEN BE THE
03600C    CORRESPONDING COLUMN OF THE INVERSE MATRIX.
03610      M2=-NMODE
03620      DO 200 J=1,N
03630      M2=M2+1

```

```

03640C CALCULATE Z FOR COLUMN J OF THE IDENTITY MATRIX.
03650 IL=(N*(J-1)+J-NZ(J)-1)*LSS+1
03660 CALL REPLACE(LS,Z((J-1)*LSS+1),A(IL))
03670 IF(J.EQ.N) GO TO 120
03680 JP=J+1
03690 DO 115 I=JP,N
03700 CALL ZERO(LS,Z((I-1)*LSS+1))
03710 IL=(N*(I-1)+I-NZ(I)-1)*LSS+1
03720 I1=I-1
03730 DO 110 K=J,I1
03740 IF(IABS(K-I).GE.NBAND) GO TO 110
03750 KL=(N*(K-1)+I-NZ(K)-1)*LSS+1
03760 CALL MULTS(LS,Z((I-1)*LSS+1),A(KL),Z((K-1)*LSS+1))
03770 11C CONTINUE
03780 115 CALL MULT(LS,WORK,A(IL),Z((I-1)*LSS+1))
03790 120 CONTINUE
03800 L=N
03810 DO 150 I=2,N
03820 L=L-1
03830 LP=L+1
03840 IF(L.LT.J) CALL ZERO(LS,Z((L-1)*LSS+1))
03850 DO 140 K=LP,N
03860 IF(IABS(K-L).GE.NBAND) GO TO 140
03870 KU=(N*(K-1)+L-NZ(K)-1)*LSS+1
03880 CALL MULTS(LS,Z((L-1)*LSS+1),A(KU),Z((K-1)*LSS+1))
03890 14C CONTINUE
03900 150 CONTINUE
03910 M1=-NMCDE
03920 DO 160 L=1,N
03930 M1=M1+1
03940 16C CALL LIST(M1,M2,Z((L-1)*LSS+1))
03950 200 CONTINUE
03960 RETURN
03970 END
03975C *****
03980 SUBROUTINE MULTS(N,A,B,C)
03985C *****
03990C A = A - B * C
04000 COMPLEX S,A(1),B(1),C(1)
04010 IS=0
04020 DO 1CC J=1,N
04030 DO 1CC I=1,N
04040 IS=IS+1
04050 S=0.0
04060 IR=I
04070 IC=(J-1)*N+1
04080 DO 50 K=1,N
04090 S=S+B(IR)*C(IC)
04100 IR=IR+N
04110 IC=IC+1
04120 50 A(IS)=A(IS)-S
04130 1CC CONTINUE

```


Figure 1 shows a 10x10 grid of 100 characters. The characters are '0', '1', 'I', 'N', 'V', and 'A'. The grid is organized into five vertical columns. Column 1 contains '0's and '1's. Column 2 contains '0's and '1's. Column 3 contains 'I's and 'T's. Column 4 contains 'I's and 'N's. Column 5 contains 'N's and 'V's. Column 6 contains 'V's and 'A's. The characters are arranged in a way that they form a large, stylized '0' shape in the center of the grid.

```

CC10CORTI,P3C,T2C,CM24CCCO,STCX3.
CC11OACCOUNT,M0326S,ROT2A.
CC12CANNERS(OUTPUT)*J. OUTNAM*DEPT 220*RLD 117-4**
CC130ATTACH(INFIL1=YSSX)
CC131ATTACH(INFIL2=ZSWX)
CC132ATTACH(INFIL4=ZWXX)
CC133ATTACH(INFIL3=ZCWX)
CC134GOTO,DEFINE.
CC135EXIT.
CC136DEFINE(OUTFIL=YSSWX)
CC14CFIN(R=G,OPT=2)
CC17LDSET(PRESET=INDEF)
CC18CLGC.
CC220/EOP
CC225      PROGRAM BOTINVA(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
CC226      1 INFIL1,TAPE1=INFIL1,INFIL2,TAPE2=INFIL2,
CC228      2 INFIL4,TAPE4=INFIL4,OUTFIL,TAPE7=OUTFIL,
CC23C      3 INFIL3,TAPE3=INFIL3)
CC232C
CC233C      BOTINVA INVERTS THE ROT 7 MATRIX WITH WIRE AND/OR CAP ADDITIONS.
CC234C
CC235C      UNIT 5 IS THE CARD READER.
CC236C      UNIT 6 IS THE LINE PRINTER.
CC238C      UNIT 7 IS A DISK FILE FOR OUTPUT OF THE Y MATRIX.
CC239C      UNIT 1 IS A DISK FILE CONTAINING THE PT MATRIX.
CC240C      UNIT 2 IS A DISK FILE CONTAINING THE O MATRIX.
CC241C      UNIT 3 IS A DISK FILE CONTAINING PART OF THE C MATRIX.
CC243C      UNIT 4 IS A DISK FILE CONTAINING THE S MATRIX.
CC244C
CC245      COMPLEX PI(12544),C(5376),Q(5376),S(2304),YI(2304)
CC250      COMPLEX W1(115),W2(48)
CC252      COMPLEX LOAD(48)
CC254      DIMENSION WGT(48)
CC26C      DIMENSION LAD(1)
CC27C      COMMON /ORDER/ MORD(2)
CC285      NFOUND=C
CC290      READ(1) NMODE,NP,NC,NPR,NF,NW,NPW,NJ
CC292      READ(1) MORD
CC294      WRITE(6,2) 'OLD',NMODE,NP,NC,NPR,NE,NW,NPW,NJ,MORD
CC296      2 FORMAT(1X,A4,'Y MATRIX PARAMETERS:',/,
CC298      1 18X,'NMODE' NP NC NPR NE NW NPW NJ,/,
CC300      2 15X,'816',/,
CC302      3 7X,'MODE NO. OF ADDITIONS:',5X,'CAPS',5X,'WIRES',/,
CC304      4 28X,'219')
CC306      READ(5,1) NC1,NP1,NF1,NW1,NPW1,NJ1
CC308      1 FORMAT(6I3)
CC310      IF(NF1.NE.C) NF1=NC1
CC312      WRITE(6,3) NC1,NP1,NF1,NW1,NPW1,NJ1
CC314      3 FORMAT(/,'NEW ADDITIONS:',/,
CC316      1 18X,'NC1' NP1 NE1 NW1 NPW1 NJ1,/,
CC318      2 14X,'816')
CC320      NMODE=2+NMODE-1

```

```

CC322      LS=NP-3
CC324      NM=LS/2
CC326      LC=NC*NM*(NPR-3)/2
CC327      LE=NE*NM
CC328      LW=(NPW-3*NW)/2
CC330      IERA=C
CC332      IERS=0
CC334      IF(NC1.EQ.0) GO TO 20
00440C CAPS ADDED.
00450      WRITE(6,8)
00452 8     FORMAT(//,1X,'CAPS ADDED TO MATRIX',//)
CC454      NC=NC1
CC456      NPR=NPR1
CC458      NE=NE1
CC460      LC=NC*NM*(NPR-3)/2
CC461      LE=NE*NM
CC462      MORD(1)=NMODE
CC463      IF(MORD(2).NE.0) MORD(1)=NMODE+1
CC464      NAD=C
CC465      IF(MORD(2).NE.0) NAD=1
CC466      LAD(1)=LW+NJ
CC468      LAD1=2*(LC+NE*NM)
CC470      IF(NW1.NE.0) IERA=1
CC471      READ(2) NP1,NC1,NPR1,NE1
CC472      IF(NP.NE.NP1) IERS=1
CC473      IF(NC.NE.NC1) IERS=2
CC474      IF(NPR.NE.NPR1) IERS=3
CC475      IF(NE.NE.NE1) IERS=4
CC476      READ(4) NC1,NPR1,NE1
CC477      IF(NC.NE.NC1) IERS=5
CC478      IF(NPR.NE.NPR1) IERS=6
CC479      IF(NE.NE.NE1) IERS=7
CC480      IF(MORD(2).EQ.0) GO TO 15
CC481      READ(3) NC1,NPR1,NE1,NW1,NPW1,NJ1
CC482      IF(NC.NE.NC1) IERS=8
CC483      IF(NPR.NE.NPR1) IERS=9
CC484      IF(NE.NE.NE1) IERS=10
CC485      IF(NW.NE.NW1) IERS=11
CC486      IF(NPW.NE.NPW1) IERS=12
CC487      IF(NJ.NE.NJ1) IERS=13
CC488 15    CONTINUE
CC490C READ SURFACE IMPEDANCE LOADING.
CC492      READ(5,11) (W2(I),I=1,NM)
CC494 11     FORMAT(1GF8.4)
CC496      NM1=NM+1
CC498      READ(5,11) (W2(I),I=NM1,LS)
CC500      WRITE(6,12)
CC502 12     FORMAT(//,1X,T AND Z SURFACE IMPEDANCE LOADING (COMPLEX))
CC504      WRITE(6,59) (W2(I),I=1,NM)
CC506 59     FORMAT(//,1X,10FR.4)
CC508      WRITE(6,59) (W2(I),I=NM1,LS)
CC510      LR=(NPR-3)/2

```

```

00528      J=0
00529      DO 100 JC=1,NC
00530      DO 100 JP=1,NM
00531      DO 100 JR=1,LR
00532      J=J+1
00533      LOAD(J)=W2(JP)
00534      100 LOAD(J+LC)=W2(JP+NM)
00535      IF(ME.EQ.0) GO TO 120
00536      J=2*LC
00537      DO 110 JC=1,NC
00538      DO 110 JP=1,NM
00539      J=J+1
00540      110 LOAD(J)=W2(JP+NM)
00541      120 CONTINUE
00542      I1=NP*LP
00543      READ(4) (WGHT(I),I=1,I1)
00544      J=0
00545      DO 130 JC=1,NC
00546      DO 130 JP=1,NM
00547      DO 130 JR=1,LR
00548      J=J+1
00549      I=I+1
00550      130 LOAD(J)=LOAD(J)*WGHT(I)
00551      READ(4) (WGHT(I),I=1,I1)
00552      DO 140 JC=1,NC
00553      I=0
00554      DO 140 JP=1,NM
00555      DO 140 JR=1,LR
00556      J=J+1
00557      I=I+1
00558      140 LOAD(J)=LOAD(J)*WGHT(I)
00559      IF(ME.EQ.0) GO TO 160
00560      READ(4) (WGHT(I),I=1,NM)
00561      DO 150 JC=1,NC
00562      I=0
00563      DO 150 JP=1,NM
00564      J=J+1
00565      I=I+1
00566      150 LOAD(J)=LOAD(J)*WGHT(I)
00567      160 CONTINUE
00568      GO TO 50
00569      20 WIRES ADDED.
00570      WRITE(6,9)
00571      9 FORMAT(//,1X,'WIRES ADDED TO MATRIX',//)
00572      NW=NW1
00573      NPW=NPW1
00574      NJ=NJ1
00575      LW=(NPW-3*NW)/2
00576      MORD(2)=NMODE
00577      IF(MCRD(1).NE.0) MORD(2)=NM7DF+1
00578      NAD=C

```

```

C0886 IF(MCRD(1).NE.0) NAD=1
C0900 LAD(1)=2*LC+NE*NM
C0910 LAD1=LW+NJ
C0920 IF(MC1.NE.0) IERA=1
C0930 READ(2) NP1,NW1,NPW1,NJ1
C0940 IF(NP.NE.NP1) IERS=1
C0950 IF(NW.NE.NW1) IERS=2
C0960 IF(NPW.NE.NPW1) IERS=3
C0970 IF(NJ.NE.NJ1) IERS=4
C0980 READ(4) NW1,NPW1,NJ1
C0990 IF(NW.NE.NW1) IERS=5
C1000 IF(NPW.NE.NPW1) IERS=6
C1010 IF(NJ.NE.NJ1) IERS=7
C1020 IF(MCRD(1).EQ.0) GO TO 50
C1030 READ(3) MC1,NP1,NF1,NW1,NPW1,NJ1
C1040 IF(MC.NE.MC1) IERS=8
C1050 IF(NP.NE.NP1) IERS=9
C1060 IF(NF.NE.NF1) IERS=10
C1070 IF(NW.NE.NW1) IERS=11
C1080 IF(NPW.NE.NPW1) IERS=12
C1090 IF(NJ.NE.NJ1) IERS=13
C1100 50 CONTINUE
C1110 WRITE(6,2) 'NEW',NMODE,NP,NC,NPP,NE,NW,NPW,NJ,MORD
C1120 IF(IERA.NE.0) GO TO 900
C1130 IF(IERS.NE.0) GO TO 900
C1140 NEXP=(KMODE+NAD)**2+KMODE+NAD+1
C1150 LL=KMODE*LS
C1160 IF(NAD.EQ.0) GO TO 250
C1170 DO 200 J=1,NAD
C1180 200 LL=LL+LAD(1)
C1190 250 CONTINUE
C1200 C DETERMINE MINIMUM DIMENSIONS.
C1210 K1=LL*LL
C1220 K2=LL*LAD1
C1230 K3=K2
C1240 K4=LAD1*LAD1
C1250 K5=MAX0(LS*LS,LS*(LW+NJ),LS*(2*LC+LE),(LW+NJ)**2,
C1260 1 (2*LC+LE)**2,(LW+NJ)*(2*LC+LE))
C1270 K6=MAX0(LL,LAD1)
C1280 K7=LAD1
C1290 WRITE(6,5) K1,K2,K3,K4,K5,K6,K7
C1300 5 FORMAT(// 'MINIMUM PROGRAM DIMENSIONS ARE AS FOLLOWS: ',
C1310 1 ' PI O P S YI W1',
C1320 2 ' W2,LOADWEIGHT',/,7I8)
C1330 CALL GETPORS(NMODE,NAD,LS,LAD,LAD1,W2,PI,C,R,S,YI,NFOUND)
C1340 IF(NFOUND.NE.NEXP) GO TO 490
C1350 IF(MCRD(1).LT.MORD(2)) GO TO 112
C1360 WRITE(6,113)
C1370 113 FORMAT(// ' CAP IMPEDANCE ADDED')
C1380 ADD CAP IMPEDANCE TO DIAGONAL.
C1390 I=1
C1400 DO 111 J=1,LAD1

```

```

C1410 S(I)=S(I)+LRAD(J)
C1420 111 I=I+LAD1+1
C1430 112 CONTINUE
C1440C INVERT THE Z MATRIX.
C1450 CALL INVPAR(P1,Q,P,S,W1,W2,LL,LAD1)
C1460C OUTPUT Z MATRIX BY MODE NUMBERS.
C1470 WRITE(7) NMDF,ND,AC,NDF,NF,NW,NPW,NJ
C1480 WRITE(7) MOPD
C1490 CALL PUTPODS(NMODE,NAD,LS,LAD,LAD1,W2,P1,Q,P,S,YI)
C1500 GO TO 500
C1510 490 WRITE(6,7)
C1520 7 FORMAT(//,41H ERROR IN THE NUMBER OF SUBMATRICES FOUND)
C1530 500 CONTINUE
C1540 WRITE(6,4) NEXP,NFOUND
C1550 4 FORMAT(//,33H NUMBER OF SUBMATRICES EXPECTED =,I5,/,
C1560 1 30H NUMBER OF SUBMATRICES FOUND =,I9)
C1570 STOP
C1580C ERROR IN DATA INPUT.
C1590 900 CONTINUE
C1600 IF(IERA.NE.0) WRITE(6,901)
C1610 901 FORMAT(//,1 *** ERROR *** BOTH CAPS AND WIRES CAN NOT RE',
C1620 1 ' ADDED AT THE SAME TIME)
C1630 IF(IERS.NE.0) WRITE(6,902) IERS
C1640 902 FORMAT(//,1 *** ERROR *** ONE OR MORE OF THE INPUT ',
C1650 1 ' PARAMETERS DO NOT AGREE WITH Z MATRIX FILES',
C1660 2 3X,'CODE =',I4)
C1670 STOP
C1680 END
C1690C *****
C1700 SUBROUTINE LIST(M,N,7I,LI,LJ)
C1710C *****
C1720C PRINT Z(M,N) ON THE LINE PRINTER AND WRITE TO DISK FILE.
C1730 DIMENSION JK(4)
C1740 COMPLEX ZI(1)
C1750 NI=LI/2
C1760 NJ=LJ/2
C1770 LSS=LI*LJ
C1780 IF(M.NE.NI) GO TO 200
C1790 JK(1)=1
C1800 JK(2)=NI+1
C1810 JK(3)=2*NI+NJ+1
C1820 JK(4)=JK(3)+NI
C1830 WRITE(6,2) M,N
C1840 2 FORMAT(4H1 M=,I3,4H N=,I3,/)
C1850 DO 140 J=1,4
C1860 K1=JK(J)
C1870 WRITE(6,3) J
C1880 3 FORMAT(2H Y,1)
C1890 DO 140 I=1,NJ
C1900 K2=K1+NJ+1
C1910 WRITE(6,4) (ZI(K),K=K1,K2)
C1920 4 FORMAT(1X,1,G11,4)

```

```

C1930      KI=KI+LI
C1940      14C CONTINUE
C1950      2CC CONTINUE
C1960      WRITE(7) M,N
C1970      WRITE(7) (ZI(I),I=J,LSS)
C1980      RETURN
C1990      END
C2000C -----
C2010C      SUBROUTINE INVPAP(PI,O,P,S,W,LP,N,M)
C2020C
C2030C      WRITTEN 8/8/80 BY JOHN M. PUTNAM DEPT 220
C2040C
C2050C      INVERT THE PARTITIONED MATRIX  $\begin{pmatrix} P & O \\ O & S \end{pmatrix}$  WHERE P INVERSE(PI) HAS
C2060C      ALREADY BEEN COMPUTED. N AND M ARE THE ORDER OF P AND S, RESPECTIVELY.
C2070C      ON RETURN, THESE ARRAYS CONTAIN THE INVERSE.
C2080C      W AND LP ARE WORK ARRAYS OF DIMENSION MAX(N,M) AND M, RESPECTIVELY.
C2090C
C2100C      COMPLEX PI(1),O(1),P(1),S(1),W(1)
C2110C      DIMENSION LP(1)
C2120C      CALL ASO(R,PI,W,M,N,+1)
C2130C      CALL ARC(S,P,C,M,N)
C2140C      CALL LINEQ(M,S,LP)
C2150C      CALL SOA(PI,O,P,N,W,+1)
C2160C      CALL ASC(O,S,W,N,W,-1)
C2170C      CALL ARC(PI,O,P,N,M)
C2180C      CALL SOA(S,P,W,M,N,-1)
C2190C      RETURN
C2200C      END
C2210C -----
C2220C      SUBROUTINE SCA(SO,A,W,N,M,I)
C2230C      FORM THE PRODUCT  $I+SO+A$ , WHERE SO IS A SQUARE MATRIX OF ORDER N.
C2240C      THE RESULT IS RETURNED IN A.
C2250C      COMPLEX SO(1),A(1),W(1),WW
C2260C      DO 400 JM=1,M
C2270C      DO 200 JN=1,N
C2280C      IA=(JM-1)*N+1
C2290C      ISO=IA
C2300C      WW=C
C2310C      DO 100 JN=1,N
C2320C      WB=WW+SO(ISO)*A(IA)
C2330C      ISO=ISO+N
C2340C      IA=IA+1
C2350C      100 CONTINUE
C2360C      W(IN)=I*WW
C2370C      200 CONTINUE
C2380C      IA=(JM-1)*N+1
C2390C      DO 300 JN=1,N
C2400C      A(IA)=W(IN)
C2410C      IA=IA+1
C2420C      300 CONTINUE
C2430C      400 CONTINUE
C2440C

```

```

C2450      RETURN
C2460      END
-----
C24700     SUBROUTINE ASO(A,SC,W,M,N,I)
C24800     FORM THE PRODUCT TAA*SC, WHERE SC IS A SQUARE MATRIX OF ORDER N.
C24900     THE RESULT IS RETURNED IN A.
C25000     COMPLEX A(1),SC(1),W(1),WW
C2510      DO 400 JM=1,M
C2520      DO 200 JN=1,N
C2530      IA=JM
C2540      ISC=(JN-1)*N+1
C2550      WW=0.
C2560      DO 100 IN=1,N
C2570      WW=WW+SC(ISC)*A(IA)
C2580      ISC=ISC+1
C2590      IA=IA+N
C2600      100 CONTINUE
C2610      W(JN)=I*WW
C2620      200 CONTINUE
C2630      IA=JM
C2640      DO 300 JN=1,N
C2650      A(IA)=W(JN)
C2660      IA=IA+N
C2670      300 CONTINUE
C2680      400 CONTINUE
C2690      RETURN
C2700      END
-----
C27100     SUBROUTINE ABC(A,B,C,N,M)
C27200     COMPUTE A = A - B * C, WHERE A IS A SQUARE MATRIX OF ORDER N.
C27300     COMPLEX A(1),B(1),C(1),WW
C27400     IA=1
C2750      DO 400 JN=1,N
C2760      DO 200 IN=1,N
C2770      IP=IN
C2780      IC=(JN-1)*N+1
C2790      WW=0.
C2800      DO 200 KP=1,N
C2810      WW=WW+B(IC)*C(KP)
C2820      IP=IP+N
C2830      IC=IC+1
C2840      200 CONTINUE
C2850      IA=IA+1
C2860      A(IA)=A(IA)-WW
C2870      300 CONTINUE
C2880      400 CONTINUE
C2890      RETURN
C2900      END
-----
C29100     SUBROUTINE LINE(ELL,C,LL)
C29200     COMPLEX MATRIX INVERSION ROUTINE.
C29300     COMPLEX C(1),SIO,SIO,SIO,SIO

```

```

02970      DIMENSION LR(1)
02980      DO 20 I=1,LL
02990      LR(I)=I
03000 20 CONTINUE
03010      M1=0
03020      DO 18 M=1,LL
03030      K=M
03040      K2=M1+K
03050      S1=CABS(C(K2))
03060      DO 2 I=M,LL
03070      K1=M1+I
03080      S2=CABS(C(K1))
03090      IF(S2-S1) 2,2,6
03100 6 K=I
03110      S1=S2
03120 2 CONTINUE
03130      LS=LR(M)
03140      LR(M)=LR(K)
03150      LR(K)=LS
03160      K2=M1+K
03170      STOR=C(K2)
03180      J1=0
03190      DO 7 J=1,LL
03200      K1=J1+K
03210      K2=J1+M
03220      STOR=C(K1)
03230      C(K1)=C(K2)
03240      C(K2)=STOR/STOR
03250      J1=J1+1
03260 7 CONTINUE
03270      K1=M1+M
03280      C(K1)=1./STOR
03290      DO 11 I=1,LL
03300      IF(I-M) 12,11,12
03310 12 K1=M1+I
03320      ST=C(K1)
03330      C(K1)=0.
03340      J1=0
03350      DO 10 J=1,LL
03360      K1=J1+I
03370      K2=J1+M
03380      C(K1)=C(K1)-C(K2)*ST
03390      J1=J1+1
03400 10 CONTINUE
03410 11 CONTINUE
03420      M1=M1+1
03430 18 CONTINUE
03440      J1=0
03450      DO 9 J=1,LL
03460      IF(J-LR(J)) 14,8,14
03470 14 LRJ=LR(J)
03480      J2=(LRJ-1)*LL

```

```

C349C 21 DO 13 I=1,LL
C350C K2=J2+I
C351C K1=J1+I
C352C S=C(K2)
C353C C(K2)=C(K1)
C354C C(K1)=S
C355C 13 CONTINUE
C356C LR(J)=LR(LRJ)
C357C LR(LRJ)=LRJ
C358C IF(J-LR(J)) 14,9,14
C359C 8 J1=J1+LL
C360C 9 CONTINUE
C361C RETURN
C362C END
C363C -----
C364C SUBROUTINE GETPQRS(NMODE,NAD,LS,LAD,LAD1,LWORK,P,Q,R,S,YI,NF)
C365C DIMENSION LAD(1),LWORK(1)
C366C COMPLEX P(1),Q(1),R(1),S(1),YI(1)
C367C COMMON /ORDER/ MORD(2)
C368C KMODE=2*NMODE-1
C369C LL=KMODE*LS
C370C LSS=LS*LS
C371C IF(NAD.EQ.0) GO TO 40
C372C DO 2C I=1,NAD
C373C LWORK(I)=LL
C374C 2C LL=LL+LAD(I)
C375C 4C CONTINUE
C376C RETRIEVE MATRIX P BY SUBMATRICES.
C377C 100 READ(1) M,N
C378C IF(EOF(1)) 200,110
C379C 110 CONTINUE
C380C NF=NF+1
C381C IF(M.GE.NMODE) GO TO 130
C382C IF(N.GE.NMODE) GO TO 150
C383C READ(1) (YI(I),I=1,LSS)
C384C II=0
C385C K=(N+NMODE-1)*LL*LS+(M+NMODE-1)*LS
C386C DO 12C J=1,LS
C387C DC 115 I=1,LS
C388C II=II+1
C389C K=K+1
C390C P(K)=YI(II)
C391C 115 CONTINUE
C392C K=K+LL-LS
C393C 12C CONTINUE
C394C GO TO 100
C395C 130 CONTINUE
C396C IF(N.GE.NMODE) GO TO 170
C397C LI=LAD(M-NMODE+1)
C398C LIJ=LI*LS
C399C READ(1) (YI(I),I=1,LIJ)
C400C II=0

```

```

C4010      K=(N+NM0DE-1)*LL*LS+LW0RK(M-NM0DE+1)
C4020      DO 140 J=1,LS
C4030      DO 135 I=1,LI
C4040      II=II+1
C4050      K=K+1
C4060      P(K)=YI(II)
C4070 135  CONTINUE
C4080      K=K+LI-LI
C4090 140  CONTINUE
C4100      GO TO 100
C4110 150  CONTINUE
C4120      LJ=LAC(N-NM0DE+1)
C4130      LIJ=LS*LJ
C4140      READ(1) (YI(I),I=1,LIJ)
C4150      II=0
C4160      K=LW0RK(N-NM0DE+1)*LL+(M+NM0DE-1)*LS
C4170      DO 160 J=1,LJ
C4180      DO 155 I=1,LS
C4190      II=II+1
C4200      K=K+1
C4210      P(K)=YI(II)
C4220 155  CONTINUE
C4230      K=K+LI-LS
C4240 160  CONTINUE
C4250      GO TO 100
C4260 170  CONTINUE
C4270      LI=LAD(M-NM0DE+1)
C4280      LJ=LAD(N-NM0DE+1)
C4290      LIJ=LI*LJ
C4300      READ(1) (YI(I),I=1,LIJ)
C4310      II=0
C4320      K=LW0RK(N-NM0DE+1)*LL+LW0RK(M-NM0DE+1)
C4330      DO 180 J=1,LJ
C4340      DO 175 I=1,LI
C4350      II=II+1
C4360      K=K+1
C4370      P(K)=YI(II)
C4380 175  CONTINUE
C4390      K=K+LI-LI
C4400 180  CONTINUE
C4410      GO TO 100
C4420 RETRIEVE R & Q BY SUBMATRICES, USING THE SYMMETRY RELATION
C4430 BETWEEN R AND Q.  Q(N) = Q(-N) TRANSPOSE.
C4440 200  CONTINUE
C4450 300  READ(2) M
C4460      IF(EOF(2)) 400,310
C4470 310  CONTINUE
C4480      LIJ=LS*LAD1
C4490      READ(2) (YI(I),I=1,LIJ)
C4500      IF(IARS(M).GE.NM0DE) GO TO 300
C4510      NF=NF+1
C4520      II=0

```

```

04530      K=(M+NM0DE-1)*LS
04540      DO 320 J=1,LAD1
04550      KT=(-M+NM0DE-1)*LAD1+LS+J
04560      DO 315 I=1,LS
04570      II=II+1
04580      K=K+1
04590      Q(K)=YI(II)
04600      R(KT)=YI(II)
04610      KT=KT+LAD1
04620 315  CONTINUE
04630      K=K+(L-LS)
04640 320  CONTINUE
04650      GO TO 300
04670C RETR IEVE ADDITIONAL PART OF MATRIX Q.
04680 400  CONTINUE
04690      IF(LAD(1).EQ.0) GO TO 500
04700      LIJ=LAD(1)+LAD1
04710      READ(3) (YI(I),I=1,LIJ)
04720      NF=NF+1
04730      DO 420 J=1,LAD1
04740      LADI=LAD(1)
04750      DO 415 I=1,LADI
04760      K=(J-1)*(KM0DE+LS+LADI)+KM0DE+LS+I
04770      KT=KM0DE+LS+LADI+(I-1)*LADI+J
04780      II=(J-1)*LADI+I
04790      IF(MORD(1).NE.NM0DE) II=(I-1)*LADI+J
04800      Q(K)=YI(II)
04810      R(KT)=YI(II)
04820 415  CONTINUE
04830 420  CONTINUE
04840C RETR IEVE MATRIX S.
04850 500  LL2=LAD1+LAD1
04860      READ(4) (S(I),I=1,LL2)
04870      NF=NF+1
04880      RETURN
04890      END
04900C -----
04910      SUBROUTINE PUTPORS(NM0DE,NAD,LS,LAD,LAD1,LWORK,P,Q,R,S,YI)
04920      DIMENSION LAD(1),LWORK(1)
04930      COMPLEX P(1),Q(1),P(1),S(1),YI(1)
04940      KM0DE=2*NM0DE-1
04950      LL=KM0DE+LS
04960      IF(NAD.EQ.0) GO TO 40
04970      DO 20 I=1,NAD
04980      LWORK(I)=LL
04990 20  LL=LL+LAC(I)
05000 40  CONTINUE
05010C STORE MATRIX P BY SUBMATRICES.
05020      DO 100 MM=1,KM0DE
05030      N=-NM0DE+MM
05040      DO 100 MM=1,KM0DE
05050      M=-NM0DE+MM

```

```

05060      II=C
05070      K=(N+NM0DE-1)*LL*LS+(M+NM0DE-1)*LS
05080      DO 8C J=1,LS
05090      DO 6C I=1,LS
05100      II=II+1
05110      K=K+1
05120      YI(II)=P(K)
05130 6C CONTINUE
05140      K=K+LL-LS
05150 8C CONTINUE
05160      CALL LIST(M,N,YI,LS,LS)
05170 1C CONTINUE
05180      IF(NAD.EQ.0) G7 TO 420
05190      DO 2CC NN=1,KM0DE
05200      N=-NM0DE+NN
05210      DO 2CC MM=1,NAD
05220      M=-NM0DE-1+MM
05230      II=0
05240      K=(N+NM0DE-1)*LL*LS+LWORK(MM)
05250      LI=LAD(MM)
05260      DO 18C J=1,LS
05270      DO 16C I=1,LI
05280      II=II+1
05290      K=K+1
05300      YI(II)=P(K)
05310 16C CONTINUE
05320      K=K+LI-LI
05330 18C CONTINUE
05340      CALL LISTA(M,N,YI,LI,LS)
05350 2C CONTINUE
05360      DO 3CC KN=1,NAD
05370      N=-NM0DE-1+NN
05380      DO 3CC MM=1,KM0DE
05390      M=-NM0DE+MM
05400      II=0
05410      K=LWORK(NN)*LL+(M+NM0DE-1)*LS
05420      LJ=LAD(MM)
05430      DO 28C J=1,LJ
05440      DO 26C I=1,LS
05450      II=II+1
05460      K=K+1
05470      YI(II)=P(K)
05480 26C CONTINUE
05490      K=K+LL-LS
05500 28C CONTINUE
05510      CALL LISTA(M,N,YI,LS,LJ)
05520 30C CONTINUE
05530      DO 4CC KN=1,NAD
05540      N=-NM0DE-1+NN
05550      DO 4CC MM=1,NAD
05560      M=-NM0DE-1+MM
05570      II=C

```

```

C558C      K=LWCRK(NN)+LL+LWCRK(MM)
C559C      LI=LAD(MM)
C560C      LJ=LAD(NN)
C561C      DO 380 J=1,LJ
C562C      DO 36C I=1,LI
C563C      II=II+1
C564C      K=K+1
C565C      YI(II)=P(K)
C566C 36C  CONTINUE
C567C      K=K+LL-LI
C568C 38C  CONTINUE
C569C      CALL LISTA(M,N,YI,LI,LJ)
C570C 4CC  CONTINUE
C571C 42C  CONTINUE
C572C  STORE MATRIX R BY SUBMATRICES.
C573C      M=NMODE+NAD
C574C      DO 5CC NN=1,KMODE
C575C      N=NMODE+NN
C576C      K=(N+NMODE-1)*LAD1+LS
C577C      CALL LISTA(M,N,P(K+1),LAD1,LS)
C578C 5CC  CONTINUE
C579C      IF(NAD.EQ.C) GO TO 610
C580C      DO 6CC NN=1,NAD
C581C      N=NMODE-1+NN
C582C      K=LWCRK(NN)+LAD1
C583C      CALL LISTA(M,N,R(K+1),LAD1,LAD(NN))
C584C 6CC  CONTINUE
C585C 61C  CONTINUE
C586C  STORE MATRIX O BY SUBMATRICES.
C587C      N=NMODE+NAD
C588C      DO 7CC MM=1,KMODE
C589C      M=NMODE+MM
C590C      II=0
C591C      K=(M+NMODE-1)*LS
C592C      LJ=LAD1
C593C      DO 68C J=1,LJ
C594C      DO 66C I=1,LS
C595C      II=II+1
C596C      K=K+1
C597C      YI(II)=O(K)
C598C 66C  CONTINUE
C599C      K=K+LL-LS
C600C 68C  CONTINUE
C601C      CALL LISTA(M,N,YI,LS,LJ)
C602C 7CC  CONTINUE
C603C      IF(NAD.EQ.C) GO TO 810
C604C      DO 8CC MM=1,NAD
C605C      M=NMODE-1+MM
C606C      II=0
C607C      K=LWCRK(MM)
C608C      LI=LAD(MM)
C609C      LJ=LAD1

```

```

C6100      DO 780 J=1,LJ
C6110      DO 760 I=1,LI
C6120      II=II+1
C6130      K=K+1
C6140      YI(II)=C(K)
C6150 760   CONTINUE
C6160      K=K+LI-LI
C6170 780   CONTINUE
C6180      CALL LISTA(M,N,YI,LI,LJ)
C6190 800   CONTINUE
C6200 810   CONTINUE
C6210C STOP E MATRIX S.
C6220      M=NMCDI+NAD
C6230      N=NMCDI+NAD
C6240      CALL LISTA(M,N,S,LA01,LA01)
C6250      RETURN
C6260      END
C6270C *****
C6280      SUBROUTINE LISTA(M,N,7I,LI,LJ)
C6290C *****
C6300C PRINT 2(M,N) ON THE LINE PRINTER AND WRITE TO DISK FILE.
C6310      COMPLEX ZI(1)
C6320      LSS=LI*LJ
C6330      IF(M.NE.N) GO TO 200
C6340      WRITE(6,2) M,N
C6350 2      FORMAT(4H1 M=,I3,4H N=,I3,/)
C6360      K1=1
C6370      WRITE(6,3)
C6380 3      FORMAT(2H Y)
C6390      DO 140 I=1,LJ
C6400      K2=K1+LI-1
C6410      WRITE(6,4) (ZI(K),K=K1,K2)
C6420 4      FORMAT(1X,10G11.4)
C6430      K1=K1+LI
C6440 140   CONTINUE
C6450 200   CONTINUE
C6460      WRITE(7) M,N
C6470      WRITE(7) (7I(I),I=1,LSS)
C6480      RETURN
C6490      END
C6500C /FOR
C6510      C      1 9 1
C6520      .0000 .0000 .0000 .0000 .0000 .0000 .0000 .0000 .0000
C6530      .0000 .0000 .0000 .0000 .0000 .0000 .0000 .0000 .0000
C6540      .0000 .0000 .0000 .0000 .0000 .0000 .0000 .0000 .0000
C6550      .0000 .0000 .0000 .0000 .0000 .0000 .0000 .0000 .0000
C6560C /FOR

```

PRECEDING PAGE ~~XXXXXXXXXX~~ ~~XXXXXXXXXX~~

```

G010CBOTA,P30,T1C,CM15C00,STCX3.
G0110ACCOUNT,M0326S,ROT2A.
G0120BANNERS(OUTPUT)*J. PUTNAM*DEPT 220*BLD 110-4**
G0130ATTACH(INFIL=YSSX)
G0140FTN(P=0,DPT=1)
G0164LDSET(PRESET=INDEF)
G017CLGC.
G0190/EOR
G0200      PROGRAM BOTAC(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
G0210      1 INFIL,TAPE1=INFIL)
G0220C
G0230C      BOTAC IS THE BOT ANTENNA COUPLING CODE.
G0240C
G0250C      UNIT 5 IS THE CARD READER.
G0260C      UNIT 6 IS THE LINE PRINTER.
G0270C      UNIT 1 IS A DISK FILE CONTAINING THE Y MATRIX.
G0280C
G0285      COMPLEX U,A3,A4
G0295      COMPLEX TAB(400),ZAB(400),WAR(100)
G0300      COMPLEX TCDEFF,ZCDEFF
G0305      COMPLEX YTT,YZZ
G0310      COMPLEX Y(2304)
G0320      COMMON /PLOT1/ NPLCT,XPLOT(200),YPLT(200),ZPLOT(200),ISYM(200)
G0325      COMMON /WAVE/ RK
G0330      COMMON /BOT1/ NMODE,NPT,NRAND
G0340      COMMON /BOT2/ NR,RL,YR(83),XR(83),Y91(82),X81(82)
G0350      COMMON /BOT3/ Q4(82),SV(82),CV(82)
G0360      COMMON /BOT5/ T(160),TP(160),TZ(160)
G0370      COMMON /BOT6/ IEDGE,IUNIF
G0380      COMMON /SLOT1/ NSA,IS(20),ZO(20),ZI(20)
G0385      COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
G0390      1 XW1(100),YW1(100),ZW1(100)
G0395      COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
G0400      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
G0410      DIMENSION MORD(2)
G0420      NPLOT=0
G0430      U=(0.,1.)
G0440      ETA=376.737
G0450      PI=3.14159265
G0460      READ(5,1) RK
G0465      1 FORMAT(E15.7)
G0470      WRITE(6,2) RK
G0475      2 FORMAT(9H1      BK,/,E15.7)
G0480C
G0490C      CALL BOTIN
G0500C
G0510C      CALL CAPIN
G0520C
G0530C      CALL WIREIN
G0540C
G0550C      CALL SLOTIN

```

```

CC710      LS=AP-3
CC712      NM=LS/2
CC720      LSS=LS+LS
CC730      LWJ=LW+AJ
CC740      LSW=LS+LWJ
CC770      LWW=LWJ**2
CC810      BKL=BK*EL
CC830      KMODE=2*NMODE-1
CC880      CALL ORDER(MORD)
CC890C CALCULATE ANTENNA COUPLING COEFFICIENTS.
CC891      K=0
CC892      DO 9C J=1,NSA
CC893      DC 9C I=1,NSA
CC894      K=K+1
CC895      TAB(K)=C.
CC896      ZAB(K)=C.
CC897 9C CONTINUE
CC900      NUMB=C
CC910 5C CONTINUE
CC920      READ(1) M,N
CC930      IF(EOF(1)) 37,60
CC940 6L NUMB=NUMB+1
CC950C DETERMINE SUBMATRIX TYPE.
CC960      IF(M.LT.NMODE .AND. N.LT.NMODE) GO TO 8C
CC970      IF(M.EQ.MORD(2) .AND. N.EQ.MORD(2)) GO TO 18C
CC980      READ(1) Y(1)
CC990      GO TO 5C
C1122C SLOT-SLOT ANTENNA COUPLING.
C1123 8C CONTINUE
C1124      READ(1) (Y(I),I=1,LSS)
C1125      IF(NSA.EQ.C) GO TO 50
C1158      K=0
C1160      DO 150 J=1,NSA
C1162      IB=IS(J)
C1164      DC 14C I=1,NSA
C1166      IA=IS(I)
C1168      YIT=Y(LS+(IB-1)+IA)
C1170      YZZ=Y(LS+(NM+IB-1)+NM+IA)
C1172      K=K+1
C1176      CALL MINT(M,ZC(I),71(I),A3,A4)
C1178      TAB(K)=TAB(K)+YIT*A3
C1180      ZAB(K)=ZAB(K)+YZZ*A4
C1214 14C CONTINUE
C1216 150 CONTINUE
C1218      GO TO 5C
C1264C WIRE-WIRE ANTENNA COUPLING.
C1265 18C CONTINUE
C1267      READ(1) (Y(I),I=1,LWW)
C1268      IF(NJ.EQ.O) GO TO 50
C1270      K=C
C1280      DO 21C JA=1,NJ
C1282      DO 21C JB=1,NJ

```

```

C1283      K=K+1
C1284      WAB(K)=Y(LWJ*(LW+JB-1)+LW+JA)
C1285      210  CONTINUE
C1344      GO TO 50
C1550C ALL Y SUBMATRICES HAVE BEEN READ.
C1560      37  CONTINUE
C1562      WRITE(6,114) NUM9
C1564      114  FORMAT(//,1X,I4,17H SUBMATRICES READ)
C1570C PRINT THE COUPLING COEFFICIENTS.
C1590      WRITE(6,101)
C1600      101  FORMAT(//,1  SLOT-SLOT ANTENNA COUPLING',/,
C1610      1  '  SLOT A  SLOT B  T AND Z COUPLING COEFFS. (COMPLEX)',/)
C1640      K=0
C1650      DO 185 J=1,NSA
C1651      IB=IS(J)
C1652      TCOEFF=0.0
C1653      ZCOEFF=0.0
C1654      H=-NM0DE
C1655      DO 184 MM=1,KM0DE
C1656      M=M+1
C1657      CALL MINT(M,ZC(J),Z1(J),A3,A4)
C1658      TCOEFF=TCOEFF+A3
C1659      ZCOEFF=ZCOEFF+A4
C1660      184  CONTINUE
C1661      J2=2*(IB-1)+1
C1662      J4=4*(IB-1)+1
C1663      TCOEFF=TCOEFF*(DH(J2)*T(J4)+DH(J2+1)*T(J4+1)+
C1664      1  DH(J2+2)*T(J4+2)+DH(J2+3)*T(J4+3))
C1665      ZCOEFF=ZCOEFF*(DH(J2)*TZ(J4)+DH(J2+1)*TZ(J4+1)+
C1666      1  DH(J2+2)*TZ(J4+2)+DH(J2+3)*TZ(J4+3))
C1667      DO 185 I=1,NSA
C1668      IA=IS(I)
C1669      I2=2*(IA-1)+1
C1670      I4=4*(IA-1)+1
C1671      K=K+1
C1672      TAB(K)=TAB(K)*(DH(I2)*T(I4)+DH(I2+1)*T(I4+1)+
C1673      1  DH(I2+2)*T(I4+2)+DH(I2+3)*T(I4+3))/TCOEFF
C1674      ZAB(K)=ZAB(K)*(DH(I2)*TZ(I4)+DH(I2+1)*TZ(I4+1)+
C1675      1  DH(I2+2)*TZ(I4+2)+DH(I2+3)*TZ(I4+3))/ZCOEFF
C1676      WRITE(6,202) I,J,TAB(K),ZAB(K)
C1677      202  FORMAT(18,I9,2X,4G10.3)
C1678      185  CONTINUE
C1679      WRITE(6,201)
C1680      201  FORMAT(//,1  WIRE-WIRE ANTENNA COUPLING',/,
C1681      1  '  JUNC A  JUNC R  COUPLING COEFF. (COMPLEX)',/)
C1682      K=C
C1683      DO 200 J=1,NJ
C1684      DC 200 I=1,NJ
C1685      K=K+1
C1686      WRITE(6,202) I,J,WAB(K)
C1687      200  CCNTINUE
C2012      CALL PLOT(XPLOT,YPLOT,ISYN,NPLOT,'X','Y',101,51)

```

```

C2814      CALL PLOT(ZPLOT,YPLOT,ISYM,NPLOT,'Z','Y',101,51)
C2816      CALL PLOT(ZPLOT,XPLOT,ISYM,NPLOT,'Z','X',101,51)
C2818      STOP
C2820      END
C3370C -----
C3375      SUBROUTINE ROTIN
C3380C
C3385C READ BOT COORDINATES AND COMPUTE BOT SEGMENT ARRAYS.
C3390C
C3392      COMMON /PLOT1/ NPLOT,XPLOT(200),YPLOT(200),ZPLOT(200),ISYM(200)
C3395      COMMON /BOT1/ NMODE,NPT,NRAND
C3400      COMMON /BOT2/ NP,RL,YB(83),XB(83),YB(82),XB(82)
C3405      COMMON /BOT3/ D4(R2),SV(82),CV(82)
C3410      COMMON /BOT5/ T(160),TP(160),TZ(160)
C3415      COMMON /BOT6/ IEDGE,IUNIF
C3420      READ(5,49) NMODE,NPT,NRAND
C3425      READ(5,49) NP
C3430      49  FORMAT(3I3)
C3435      WRITE(6,48) NMODE,NPT,NRAND,NP
C3440      48  FORMAT(32H NMODE NPT NRAND NP,/,4I8)
C3445      READ(5,53)(YB(I),I=1,NP)
C3450      READ(5,53)(XB(I),I=1,NP)
C3455      53  FORMAT(1CF8.4)
C3460      WRITE(6,55)
C3465      55  FORMAT(/,3H YB)
C3470      WRITE(6,46)(YB(I),I=1,NP)
C3475      46  FORMAT(1X,1CF8.4)
C3480      WRITE(6,56)
C3485      56  FORMAT(/,3H XB)
C3490      WRITE(6,46)(XB(I),I=1,NP)
C3495C PLOT THE BODY COORDINATES.
C3498      CALL PLOTB(XB,YB,NP,41)
C3500      READ(5,53) RL
C3505      WRITE(6,47) RL
C3510      47  FORMAT(/,21H HALF-LENGTH OF BOT =,F12.4)
C3515C UPDATE PLOT ARRAY.
C3520      DO 150 I=1,NP
C3525      NPLOT=NPLOT+1
C3530      XPLOT(NPLOT)=XB(I)
C3535      YPLOT(NPLOT)=YB(I)
C3540      ZPLOT(NPLOT)=RL
C3545      ISYM(NPLOT)='R'
C3550      NPLOT=NPLOT+1
C3555      XPLOT(NPLOT)=XB(I)
C3560      YPLOT(NPLOT)=YB(I)
C3565      ZPLOT(NPLOT)=-RL
C3570      ISYM(NPLOT)='R'
C3575      150 CONTINUE
C3580C CHECK IF BOT CONTAINS AN ODD NUMBER OF POINTS.
C3585      IF(MOD(NP,2).NE.1) GO TO 980
C3590C CHECK IF GENERATING CURVE IS OPEN OR CLOSED.
C3595      IEDGE=1

```

```

C3540 IF((YB(1)-YB(NP)).NE.0..OR.(XB(1)-XB(NP)).NE.0.) GO TO 1C
C3545 IEDGE=0
C3550 YB(NP+1)=YB(2)
C3555 XB(NP+1)=XB(2)
C3560 YB(NP+2)=YB(3)
C3565 XB(NP+2)=XB(3)
C3570 NP=NP+2
C3575 WRITE(6,66) NP
C3580 66 FORMAT(/,39H NOT GENERATING CURVE IS CLOSED. NP = ,I3)
C3585 10 CONTINUE
C3590C COMPUTATION OF BODY SEGMENT PARAMETERS.
C3595 DO 57 I=2,NP
C3600 I2=I-1
C3605 RR1=YB(I)-YB(I2)
C3610 RR2=XB(I)-XB(I2)
C3615 DH(I2)=SQRT(PRI*RR1+RR2*RR2)
C3620 XR1(I2)=.5*(XB(I)+XB(I2))
C3625 YR1(I2)=.5*(YB(I)+YB(I2))
C3630 SV(I2)=RR1/DH(I2)
C3635 CV(I2)=RR2/DH(I2)
C3640 57 CONTINUE
C3645C CHECK IF NOT SEGMENTATION IS UNIFORM.
C3650 IUNIF=C
C3655 NP1=NP-1
C3660 DO 60 I=2,NP1
C3665 RP1=DH(I)/DH(I1)
C3670 IF(RP1.LT.0.99..OR. RP1.GT.1.01) GO TO 2C
C3675 60 CONTINUE
C3680 IUNIF=1
C3685 WRITE(6,67)
C3690 67 FORMAT(/,39H NOT GENERATING CURVE HAS UNIFORM SEGMENTATION)
C3695 20 CONTINUE
C3700C COMPUTATION OF TRIANGLE FUNCTIONS T.
C3705 NM=(NP-3)/2
C3710 DO 74 J=1,NM
C3715 J2=2*(J-1)+1
C3720 J3=J2+1
C3725 J4=J3+1
C3730 J5=J4+1
C3735 J6=4*(J-1)+1
C3740 J7=J6+1
C3745 J8=J7+1
C3750 J9=J8+1
C3755 DEL1=DH(J2)+DH(J3)
C3760 DEL2=DH(J4)+DH(J5)
C3765 TP(J6)=1./DEL1
C3770 TP(J7)=1./DEL1
C3775 TP(J8)=-1./DEL2
C3780 TP(J9)=-1./DEL2
C3785 T(J6)=DH(J2)/2./DEL1
C3790 T(J7)=(DH(J2)+DH(J3)/2.)/DEL1
C3795 T(J8)=(DH(J4)/2.+DH(J5))/DEL2

```

```

C3800 T(J9)=DH(J5)/2./DFL2
C3805 74 CONTINUE
C3810 MM4=NM4
C3815 DO 75 J=1,MM4
C3820 75 T(J)=T(J)
C3825 IF(IEGE.EQ.0) GO TO 76
C3830 T(1)=2.-T(1)
C3835 T(2)=2.-T(2)
C3840 T(NM4-1)=2.-T(NM4-1)
C3845 T(NM4)=2.-T(NM4)
C3850 76 CONTINUE
C3855 RETURN
C3860 980 WRITE(6,981)
C3865 981 FORMAT(/, ' *** ERROR IN ROT INPUT')
C3870 STOP
C3875 END
C3880C *****
C3882 SUBROUTINE PLOT8(X,Y,N,NR)
C3884C *****
C3886C
C3888C WRITTEN 2/14/74 BY J. M. PUTNAM DEPT 22C X23877
C3890C
C3892C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
C3894C N IS THE NUMBER OF POINTS TO BE PLOTTED.
C3896C NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
C3898C NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
C3900C NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
C3902C
C3904 REAL X(1),Y(1),HEAD(10)
C3906 INTEGER LINE(10),BLANK,STAR,PLUS
C3908 DATA BLANK,STAR,PLUS /1H,14H,14H/
C3910 NC=51
C3912 N10=(NC-1)/10
C3914 WRITE(6,500)
C3916 500 FORMAT(/,44H BODY COORDINATES, + INDICATES TRIANGLE PEAK)
C3918 WRITE(6,504)
C3920 XMIN=X(1)
C3922 XMAX=X(1)
C3924 YMIN=Y(1)
C3926 YMAX=Y(1)
C3928 DO 6 I=1,N
C3930 IF(X(I).LT.XMIN) XMIN=X(I)
C3932 IF(X(I).GT.XMAX) XMAX=X(I)
C3934 IF(Y(I).LT.YMIN) YMIN=Y(I)
C3936 IF(Y(I).GT.YMAX) YMAX=Y(I)
C3938 6 CONTINUE
C3940 DEL=XMAX-XMIN
C3942 IF(YMAX-YMIN.GT.DEL) DEL=YMAX-YMIN
C3944 XMAX=XMIN+DEL
C3946 YMAX=YMIN+DEL
C3948 DO 5 I=1,N10
C3950 5 I=I

```

```

C3952      5 HEAD(I)=(XMAX-XMIN)*Z/NIC+XMIN
C3954      DY=(YMAX-YMIN)/(NP-1)
C3956      IEDGE=1
C3958      IF(X(1).EQ.X(N) .AND. Y(1).EQ.Y(N)) IEDGE=0
C3960      Z=YMAX+DY
C3962      YL=Z-DY/2.
C3964      DO 7 J=1,NP
C3966      DO 8 K=1,NC
C3968      8 LINE(K)=BLANK
C3970      Z=Z-DY
C3972      YU=YL
C3974      YL=Z-DY/2.
C3976      DO 9 I=1,N
C3978      IF(Y(I).GE.YU) GO TO 9
C3980      IF(Y(I).LT.YL) GO TO 9
C3982      K=(X(I)-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
C3984      IF(K.GT.NC) K=NC
C3986      LINE(K)=STAR
C3988      IF(MOD(I,2).EQ.1) LINE(K)=PLUS
C3990      IF(IEDGE.EQ.1) GO TO 9
C3992      IF(I.EQ.1 .OR. I.EQ.N) LINE(K)=STAR
C3994      9 CONTINUE
C3996      WRITE(6,508) 7,(LINE(K),K=1,NC)
C3998      7 CONTINUE
C4000      WRITE(6,504)
C4002      WRITE(6,3002)
C4004      WRITE(6,507) XMIN,(HEAD(I),I=1,N1)
C4006      RETURN
C4008      504 FORMAT (1X,14(1H-),14.,10(5H---.),1H- )
C4010      507 FORMAT(10X,11(1H-))
C4012      508 FORMAT (1X,F12.4,1X,1HI,51A1,1HI )
C4014      3002 FORMAT(4X,7HYH / 4X,4X,1HI,5(9X,1HI))
C4016      END
C4410C -----
C4420C      SUBROUTINE CAPTN
C4430C
C4440C      READ CAP INPUTS AND COMPUTE CAP ARRAYS.
C4450C
C4460      COMMON /PLOT1/ NPLOT,XPLNT(200),YPLNT(200),ZPLNT(200),ISYM(200)
C4470      COMMON /BOT2/ NP,BL,YR(R3),XR(R3),Y91(82),XB1(82)
C4480      COMMON /CAP1/ NC,XC,YC,ZC(2)
C4490      COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
C4500      COMMON /CAP3/ TCR(36),TC1(36),TPCR(36),TPCT(36)
C4510      COMMON /CAP4/ RC(R3),RC1(R2),AC(82),CPC(82),SPC(82)
C4520      COMMON /EDG1/ NE,ZE(2),ZRE(10)
C4530      COMMON /EDG2/ TCF(10),TPCF(10),TRF(10),TPBE(10)
C4540      READ(5,1) NC,NPR,NE
C4550      1 FORMAT(3I3)
C4560      IF(NE.NE.0) NF=NC
C4570      WRITE(6,3) NC,NPR,NE
C4580      3 FORMAT(16I1 NC,NPR NE,/,3I5,/)
C4590      IF(NC.EQ.0) RETURN

```

```

C4600 READ(5,2) XC,YC
C4610 2 FORMAT(10F8.4)
C4620 READ(5,2) (ZC(I),I=1,NC)
C4630 READ(5,2) (RHOC(I),I=1,NPR)
C4640 IF(NE.NE.0) READ(5,2) (ZE(I),I=1,NE)
C51000 UPDATE PLCT ARRAY.
C5102 DO 150 I=1,NC
C5104 NPLOT=NPLOT+1
C5106 XPLOT(NPLOT)=XC
C5108 YPLOT(NPLOT)=YC
C5110 ZPLOT(NPLOT)=ZC(I)
C5112 ISYM(NPLOT)='C'
C5114 IF(NE.EQ.0) GO TO 150
C5116 NPLOT=NPLOT+1
C5118 XPLOT(NPLOT)=XB(1)
C5120 YPLOT(NPLOT)=YB(1)
C5122 ZPLOT(NPLOT)=ZE(1)
C5124 ISYM(NPLOT)='E'
C5126 150 CONTINUE
C5144 WRITE(6,4)
C5146 4 FORMAT(37H CAP XC YC ZC ZE,/)
C5148 DO 100 I=1,NC
C5149 IF(NE.EQ.0) WRITE(6,5) I,XC,YC,ZC(I)
C5150 IF(NE.NE.0) WRITE(6,5) I,XC,YC,ZC(I),ZE(I)
C5151 100 CONTINUE
C5152 5 FORMAT(14,4X,4F8.4)
C5154 WRITE(6,6)
C5156 6 FORMAT(1/,5H RHOC)
C5158 WRITE(6,7) (RHOC(I),I=1,NPR)
C5160 7 FORMAT(1X,10F8.4)
C5162 IF(MCD(NPR,2).NE.1) GO TO 990
C5164 DO 120 I=2,NPR
C5166 IF(RHOC(I).LE.RHOC(I-1)) GO TO 980
C5168 CCMPUTATION OF CAP SECTOR PARAMETERS.
C5170 DO 47 I=1,NP
C5172 RC(I)=SQRT((YB(I)-YC)**2+(XB(I)-XC)**2)
C5174 47 CONTINUE
C5176 DO 57 I=2,NP
C5178 I2=I-1
C5180 RR1=YB(I2)-YC
C5182 RR2=XB(I2)-XC
C5184 RC1(I2)=SQRT(RR1*RR1+RR2*RR2)
C5186 AC(I2)=ABS((XB(I2)-XC)*(YB(I2)+YC)+(XB(I2)-XB(I2))*
C5188 1 (YB(I2)+YB(I2))+(XC-XB(I2))*(YC+YB(I2)))/2.
C5190 SPC(I2)=RR1/RC1(I2)
C5192 CPC(I2)=RR2/RC1(I2)
C5194 57 CONTINUE
C5196 DO 67 I=2,NPR
C5198 I2=I-1
C5200 RHOC1(I2)=(RHOC(I)+RHOC(I2))/2.
C5202 DRHOC(I2)=RHOC(I)-RHOC(I2)
C5204 67 CONTINUE

```

```

C5206C COMPUTATION OF CAP TRIANGLE FUNCTIONS.
C5208    LC=(NPR-3)/2
C5210    DO 74 J=1,LC
C5212    J2=2*(J-1)+1
C5214    J3=J2+1
C5216    J4=J3+1
C5218    J5=J4+1
C5220    J6=4*(J-1)+1
C5222    J7=J6+1
C5224    J8=J7+1
C5226    J9=J8+1
C5228    DEL1=DRHOC(J2)+DRHOC(J3)
C5230    DEL2=CPHOC(J4)+DRHOC(J5)
C5232    TPCR(J6)=1./DEL1
C5234    TPCR(J7)=1./DEL1
C5236    TPCR(J8)=-1./DEL2
C5238    TPCR(J9)=-1./DEL2
C5240    TCR(J6)=DRHOC(J2)/2./DEL1
C5242    TCR(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
C5244    TCR(J8)=(DRHOC(J4)/2.+DRHOC(J5))/DEL2
C5246    TCR(J9)=DRHOC(J5)/2./DEL2
C5248    74 CONTINUE
C5250    LC4=LC*4
C5252    DO 75 I=1,LC4
C5254    TCT(I)=TCR(I)
C5256    TPCT(I)=TPCR(I)
C5258    75 CONTINUE
C5260    TCT(LC4-1)=2.-TCT(LC4-1)
C5262    TCT(LC4)=2.-TCT(LC4)
C5264    TPCT(LC4-1)=-TPCT(LC4-1)
C5266    TPCT(LC4)=-TPCT(LC4)
C5268    IF(PHOC(1).EQ.0.0) GO TO 76
C5270    TCT(1)=2.-TCT(1)
C5272    TCT(2)=2.-TCT(2)
C5274    TPCT(1)=-TPCT(1)
C5276    TPCT(2)=-TPCT(2)
C5278    76 CONTINUE
C5279    IF(NE.EQ.0) RETURN
C5280C COMPUTATION OF EDGE HALF TRIANGLE FUNCTIONS.
C5282    DO 80 IC=1,NC
C5284    J2=NPR-2
C5286    J3=J2+1
C5288    J6=2*(IC-1)+1
C5290    J7=J6+1
C5292    DEL1=DRHOC(J2)+DRHOC(J3)
C5294    TPCF(J6)=1./DEL1
C5296    TPCF(J7)=1./DEL1
C5298    TCE(J6)=DRHOC(J2)/2./DEL1
C5300    TCE(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
C5302    DEL2=ZC(IC)-ZC(IC)
C5304    ZBE(J6)=ZC(IC)+0.25*DEL2
C5306    ZBE(J7)=ZC(IC)+0.75*DEL2

```

```

C5318      IF (DEL2.LT.0.0) GO TO 78
C5319C EDGE IS AT 7=-L.
C5312      TPBE(J6)=-1./DEL2
C5314      TPBE(J7)=-1./DEL2
C5316      TBE(J6)=C.75
C5318      TBE(J7)=C.25
C5320      GO TO 80
C5322  78 CONTINUE
C5324C EDGE IS AT 7=+L.
C5326      TPBE(J6)=1./DEL2
C5328      TPBE(J7)=1./DEL2
C5330      TBE(J6)=-0.75
C5332      TBE(J7)=-0.25
C5334  80 CONTINUE
C5336      RETURN
C5338  980 WRITE(6,931)
C5340  981 FORMAT(/,' **** ERROR IN CAP INPUT')
C5342      STOP
C5344      END
C53100 -----
C5315      SUBROUTINE WIREIN
C5320C
C5325C READ WIRE COORDINATES AND COMPUTE WIRE SEGMENT ARRAYS.
C5330C
C5332      COMMON /PLOT1/ NPLCT,YPLCT(200),YPLCT(200),ZPLCT(200),TSYN(200)
C5335      COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
C5340  1 XW1(100),YW1(100),ZW1(100)
C5345      COMMON /WIRE2/ DHW(100),DXW(100),DYW(100),DZW(100)
C5350      COMMON /WIRE3/ NW,INDW(6),RADW(5)
C5355      COMMON /WIRE4/ LW,TW(195),TPW(195),INDTW(49)
C5360      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
C5365      COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
C5370      COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
C5375      COMMON /JUNC4/ UXJ(10),UYJ(10),UZJ(10)
C5380      COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
C5385      COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
C5390      NW=0
C5395      NJ=0
C5400      LW=0
C5405      READ(5,51) NW,NPW,NJ
C5410  51 FORMAT(3I3)
C5415      WRITE(6,1) NW,NPW,NJ
C5420  1 FORMAT(24H1 NW NPW NJ,/,3I8)
C5425      IF (NW.EQ.0) RETURN
C5430      READ(5,53) (XW(I),I=1,NPW)
C5435      READ(5,53) (YW(I),I=1,NPW)
C5440      READ(5,53) (ZW(I),I=1,NPW)
C5445  53 FORMAT(1CF8,4I)
C5450      READ(5,52) (INDW(I),I=1,NW)
C5455  52 FORMAT(10I8)
C5460      INDW(NW+1)=NPW+1
C5465      READ(5,53) (RADW(I),I=1,NW)

```

```

C5675 IF (NJ.EQ.3) GO TO 50
C5675C NOTE, INDJW MUST BE MONOTONIC INCREASING.
C5680 READ(5,52) (INDJW(I),I=1,NJ)
C5685 READ(5,53) (RADJ(I),I=1,NJ)
C5690 READ(5,53) (UXJ(I),I=1,NJ)
C5695 READ(5,53) (UYJ(I),I=1,NJ)
C5700 READ(5,53) (UZJ(I),I=1,NJ)
C5705 50 IERM=C
C5710 IERJ=C
C5715 WRITE(6,61)
C5720 61 FORMAT(1,22X,'WIRE COORDINATES',20X,'JUNCTION PARAMETERS',/,
C5725 1 26X,'I',4X,'XW',6X,'YW',6X,'ZW',
C5730 2 10X,'IJ',4X,'RADJ',5X,'UXJ',5X,'UYJ',5X,'UZJ')
C5735 IJ=1
C5740C THIS LOOP LISTS WIRE/JUNCTION POINTS, WHILE CHECKING THE FOLLOWING:
C5745C 1) EACH WIRE MUST CONTAIN AN ODD NUMBER OF POINTS.
C5750C 2) EACH JUNCTION MUST EITHER START OR TERMINATE A WIRE.
C5755C 3) CHECK THAT ALL JUNCTION POINTS ARE FOUND.
C5760 DO 100 IW=1,NW
C5765 WRITE(6,62) IW,RADW(IW)
C5770 62 FORMAT(2X,'WIRE',I3,' RADW=',F8.4)
C5775 I1=INDW(IW)
C5780 I2=INDW(IW+1)-1
C5785C CHECK FOR AN ODD NUMBER OF POINTS ON WIRE IW.
C5790 IF(MOD(I2-I1+1,2).NE.1) IERM=1
C5795 DO 90 I=I1,I2
C5800 WRITE(6,63) I,Y4(I),YW(I),ZW(I)
C5805 63 FORMAT(25X,I3,3F8.4)
C5810C UPDATE PLOT ARRAY.
C5815 NPLUT=NPLUT+1
C5820 XPLUT(NPLUT)=YW(I)
C5825 YPLUT(NPLUT)=YW(I)
C5830 ZPLUT(NPLUT)=ZW(I)
C5835 ISYM(NPLUT)=IW
C5840C CHECK IF WIRE POINT I IS A JUNCTION POINT.
C5845 IF(IJ.GT.NJ) GO TO 90
C5850 IF(INDJW(IJ).NE.I) GO TO 90
C5855C CHECK THAT JUNCTION POINT IJ IS AT THE START OR END OF WIRE.
C5860 IF(I.NE.I1.AND. J.NE.I2) IERJ=1
C5865 WRITE(6,64) IJ,RADJ(IJ),UXJ(IJ),UYJ(IJ),UZJ(IJ)
C5870 64 FORMAT(14,58X,I3,4F8.4)
C5875C UPDATE PLOT ARRAY.
C5880 ISYM(NPLCT)=IJ
C5885C DETERMINE DIRECTION IN WHICH WIRE LEAVES JUNCTION POINT.
C5890 INDJ(IJ)=1
C5895 IF(I.EQ.I2) INDJ(IJ)=-1
C5900C COMPUTE JUNCTION PARAMETERS.
C5905 RADJ(IJ)=RADW(I4)
C5910 XJ(IJ)=XW(I)
C5915 YJ(IJ)=YW(I)
C5920 ZJ(IJ)=ZW(I)
C5925 IJ=IJ+1

```

```

C589C 9C CONTINUE
C5895 1C CONTINUE
C590CC CHECK FOR WIRE OR JUNCTION INPUT ERRORS.
C590C5 IF (IERW.NE.C) GO TO 98C
C5910 IF (IJ-1.NE.NJ) OR (IERJ.NE.O) GO TO 99C
C5915C COMPUTATION OF WIRE SEGMENT PARAMETERS.
C5920 DO 57 I=2,NPW
C5925 I2=I-1
C5930 DXW(I2)=XW(I)-XW(I2)
C5935 DYW(I2)=YW(I)-YW(I2)
C5940 DZW(I2)=ZW(I)-ZW(I2)
C5945 DHW(I2)=SQRT(DXW(I2)**2+DYW(I2)**2+DZW(I2)**2)
C5950 XW1(I2)=0.5*(XW(I)+XW(I2))
C5955 YW1(I2)=0.5*(YW(I)+YW(I2))
C5960 ZW1(I2)=0.5*(ZW(I)+ZW(I2))
C5965 57 CONTINUE
C597CC COMPUTATION OF WIRE TRIANGLE FUNCTIONS TW.
C5975 LW=L
C5980 DO 75 IW=1,NW
C5985 I1=INOW(IW)
C5990 I2=INOW(IW+1)-1
C5995 LW1=(I2-I1-2)/2
C600C DO 74 J=1,LW1
C6005 LW=LW+1
C6010 J2=2*(J-1)+I1
C6015 J3=J2+1
C6020 J4=J3+1
C6025 J5=J4+1
C6030 J6=4*(LW-1)+1
C6035 J7=J6+1
C6040 J8=J7+1
C6045 J9=J8+1
C605C INDTW(LW)=J2
C6055 DEL1=DHW(J2)+DHW(J3)
C606C DEL2=DHW(J4)+DHW(J5)
C6065 TPW(J6)=1./DEL1
C6070 TPW(J7)=1./DEL1
C6075 TPW(J8)=1./DEL2
C6080 TPW(J9)=1./DEL2
C6085 TW(J6)=DHW(J2)/2./DEL1
C6090 TW(J7)=(DHW(J2)+DHW(J3))/2./DEL1
C6095 TW(J8)=(DHW(J4)+DHW(J5))/2./DEL2
C610C TW(J9)=DHW(J5)/2./DEL2
C6105 74 CONTINUE
C6110 75 CONTINUE
C6115C NOTE, LW=(NPW-NW)/2-NW
C6120 IF (INJ.EQ.O) RETURN
C6125C COMPUTATION OF JUNCTION HALF TRIANGLE FUNCTIONS TJ.
C6130 DO 85 IJ=1,NJ
C6135 IF (INDTJ(IJ).GT.O) GO TO 80
C6140C JUNCTION IS AT THE END OF A WIRE.
C6145 J2=INDJW(IJ)-2

```

```

06150      J3=J2+1
06155      J6=2*(IJ-1)+1
06160      J7=J6+1
06165      IND TJ(IJ)=J2
06170      DEL1=-DHW(J2)-DHW(J3)
06175      TPJ(J6)=1./DEL1
06180      TPJ(J7)=1./DEL1
06185      TJ(J6)=DHW(J2)/2./DEL1
06190      TJ(J7)=(DHW(J2)+DHW(J3)/2.)/DEL1
06195      GO TO 85
06200C JUNCTION IS AT THE START OF A WIRE.
06205 80 J4=IND JW(IJ)
06210      J5=J4+1
06215      J8=2*(IJ-1)+1
06220      J9=J8+1
06225      IND TJ(IJ)=J4
06230      DEL2=DHW(J4)+DHW(J5)
06235      TPJ(J8)=-1./DEL2
06240      TPJ(J9)=-1./DEL2
06245      TJ(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
06250      TJ(J9)=DHW(J5)/2./DEL2
06255 85 CONTINUE
06260C COMPUTE UNIT VECTORS FOR JUNCTION(DISK).
06265      DC 95 IJ=1,NJ
06270C NORMAL UNIT VECTOR (UJ).
06275      RR=SQRT(UXJ(IJ)**2+UYJ(IJ)**2+UZJ(IJ)**2)
06280      UXJ(IJ)=UXJ(IJ)/RR
06285      UYJ(IJ)=UYJ(IJ)/RR
06290      UZJ(IJ)=UZJ(IJ)/RR
06295C FIND 2 ORTHOGONAL UNIT VECTORS IN THE PLANE OF THE DISK (UJ1 & UJ2).
06300      UXJ1(IJ)=0.0
06305      UYJ1(IJ)=0.0
06310      UZJ1(IJ)=0.0
06315      UXJ2(IJ)=0.0
06320      UYJ2(IJ)=0.0
06325      UZJ2(IJ)=0.0
06330      IF(UXJ(IJ).EQ.0.0) GO TO 91
06335C FIND INTERSECTION WITH X-Y PLANE.
06340      UYJ1(IJ)=1.0
06345      UXJ1(IJ)=-UYJ(IJ)/UXJ(IJ)
06350C FIND INTERSECTION WITH X-Z PLANE.
06355      UZJ2(IJ)=1.0
06360      UXJ2(IJ)=-UZJ(IJ)/UXJ(IJ)
06365      GO TO 94
06370 91 IF(UYJ(IJ).EQ.0.0) GO TO 92
06375C FIND INTERSECTION WITH Y-Z PLANE.
06380      UZJ1(IJ)=1.0
06385      UYJ1(IJ)=-UZJ(IJ)/UYJ(IJ)
06390C FIND INTERSECTION WITH X-Y PLANE.
06395      UXJ2(IJ)=1.0
06400      UYJ2(IJ)=-UXJ(IJ)/UYJ(IJ)
06405      GO TO 94

```

```

06410 92 IF(UZJ(IJ),EQ.0.0) GO TO 94
06415C FIND INTERSECTION WITH X-Z PLANE.
06420 UXJ1(IJ)=1.0
06425 UZJ1(IJ)=-UXJ(IJ)/UZJ(IJ)
06430C FIND INTERSECTION WITH Y-Z PLANE.
06435 UYJ2(IJ)=1.0
06440 UZJ2(IJ)=-UYJ(IJ)/UZJ(IJ)
06445 94 CONTINUE
06450 RR=SQRT(UXJ1(IJ)**2+UYJ1(IJ)**2+UZJ1(IJ)**2)
06455 UXJ1(IJ)=UXJ1(IJ)/RR
06460 UYJ1(IJ)=UYJ1(IJ)/RR
06465 UZJ1(IJ)=UZJ1(IJ)/RR
06470 RR=SQRT(UXJ2(IJ)**2+UYJ2(IJ)**2+UZJ2(IJ)**2)
06475 UXJ2(IJ)=UXJ2(IJ)/RR
06480 UYJ2(IJ)=UYJ2(IJ)/RR
06485 UZJ2(IJ)=UZJ2(IJ)/RR
06490 95 CONTINUE
06495 RETURN
06500 980 WRITE(6,981)
06505 981 FORMAT(/, ' *** ERROR IN WIRE INPUT')
06510 STOP
06515 990 WRITE(6,991)
06520 991 FORMAT(/, ' *** ERROR IN JUNCTION INPUT')
06525 STOP
06530 END
07650C -----
07660 SUBROUTINE SLOTIN
07670C
07680C READ SLOT ANTENNA INPUTS.
07690C
07700 COMMON EO
07702 COMMON /PLOT1/ NPLOT,XPLOT(200),YPLOT(200),ZPLOT(200),ISYM(200)
07704 COMMON /BOT2/ NP,BL,YB(83),XB(83),Y91(82),XB1(82)
07710 COMMON /SLOT1/ NSA,IS(20),ZO(20),Z1(20)
07720 COMMON /SLOT2/ EO(20),TEXC(20),ZEXC(20)
07730 READ(5,16) NSA
07740 16 FORMAT(I3)
07750 WRITE(6,3) NSA
07760 3 FORMAT(/,26HNUMBER OF SLOT ANTENNAS =,I3)
07770 IF(NSA.EQ.0) RETURN
07780 READ(5,54) (IS(K),K=1,NSA)
07790 54 FORMAT(10I8)
07800C NOTE -L <= ZO(K) < Z1(K) <= L
07810 READ(5,53) (ZO(K),K=1,NSA)
07820 53 FORMAT(10F8,4)
07830 READ(5,53) (Z1(K),K=1,NSA)
07840 READ(5,53) (EC(K),K=1,NSA)
07850C T-EXCITATION.
07860 READ(5,53) (TEXC(K),K=1,NSA)
07870C Z-EXCITATION.
07880 READ(5,53) (ZEXC(K),K=1,NSA)
07890 WRITE(6,4) (K,IS(K),ZO(K),Z1(K),EO(K),TEXC(K),ZEXC(K)),K=1,NSA)

```

```

07900 4   FORMAT(//,50H ANTENNA NO.   IS   Z0   Z1   EO,
07910     1 11X,14HTEXC   ZEXC,//,(I9,I8,6F10.4))
07911C UPDATE PLOT ARRAY,
07912   DO 150 I=1,NSA
07913     NPLOT=NPLOT+1
07914     IND=IS(I)
07915     XPLOT(NPLOT)=XB(2*IND+1)
07916     YPLOT(NPLOT)=YB(2*IND+1)
07917     ZPLOT(NPLOT)=Z0(I)
07918     ISYM(NPLOT)='S'
07919     NPLOT=NPLOT+1
07920     XPLOT(NPLOT)=XB(2*IND+1)
07921     YPLOT(NPLOT)=YB(2*IND+1)
07922     ZPLOT(NPLOT)=Z1(I)
07923     ISYM(NPLOT)='S'
07924 150 CONTINUE
07925 RETURN
07930 END
07940C -----
07950 SUBROUTINE ORDER(MORD)
07960C DETERMINE THE ORDER IN WHICH THE Y MATRIX WAS GENERATED, AND
07970C CHECK FOR DIMENSIONAL CONSISTENCY BETWEEN THE Y MATRIX DATA FILE,
07980C AND THE INPUT DATA FILE.
07990 DIMENSION MORD(2)
08000 COMMON /BOT1/  NMODE,NPT,NBAND
08010 COMMON /BOT2/  NP,BL,YB(83),XB(83),YB1(82),XB1(82)
08020 COMMON /CAP1/  NC,XC,YC,ZC(2)
08030 COMMON /CAP2/  NPR,PHOC(21),RHOC1(20),DRHOC(20)
08040 COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
08050 1 XW1(100),YW1(100),ZW1(100)
08060 COMMON /WIRE3/ NW,INDW(6),RADW(5)
08070 COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
08080 COMMON /EDGE1/ NE,ZE(2),7BE(10)
08090 READ(1) NMODE,NP1,NC1,NPR1,NE1,NW1,NPW1,NJ1
08100 IF(NMODE.NE.NMODE1) GO TO 500
08110 IF(NP.NE.NP1) GO TO 500
08120 IF(NC.NE.NC1) GO TO 500
08130 IF(NPR.NE.NPR1) GO TO 500
08140 IF(NE.NE.NE1) GO TO 500
08150 IF(NW.NE.NW1) GO TO 500
08160 IF(NPW.NE.NPW1) GO TO 500
08170 IF(NJ.NE.NJ1) GO TO 500
08180C MORD SPECIFIES THE ORDER IN WHICH CAPS AND/OR WIRES HAVE BEEN ADDED
08190C TO THE BOT. INDEX 1 REFERS TO CAPS AND INDEX 2 REFERS TO WIRES.
08200C MORD(I)=0 IF I HAS NOT BEEN ADDED
08210C MORD(I)=N IF I HAS BEEN ADDED TO THE SYSTEM MATRIX, AND
08220C IS LOCATED AT PSEUDO MODE NUMBER N.
08230 READ(1) MORD
08240 WRITE(6,4) MORD
08250 4   FORMAT(' THE Y MATRIX CONTAINS THE FOLLOWING ADDITIONS ( 0 IF ',
08260 1 ' NOT INCLUDED), OR CORRESPONDING MODE NUMBER IF PRESENT)',//,
08270 2 1CX,'CAPS',I4,/,1CX,'WIRES',I3)

```

```

C8260      RETURN
C8270 500  WRITE(6,1)
C8280 1    FORMAT(//,' *** ERROR ***  INPUT PARAMETERS DO NOT CHECK WITH',
C8290      1 ' THE PARAMETERS IN THE Y MATRIX FILE',//,
C8300      2 6X,' NMODE NP NC NPR NE NW NPW NJ')
C8310      WRITE(6,2) NMODE,NP,NC,NPR,NE,NW,NPW,NJ
C8320 2    FORMAT(' INPUT ',8I6)
C8330      WRITE(6,3) NMODE1,NP1,NC1,NPR1,NE1,NW1,NPW1,NJ1
C8340 3    FORMAT(' Y FILE',8I6)
C8350      STOP
C8360      END
C8370C -----
C8380C      SUBROUTINE MINT(M,Z0,Z1,TINT,ZINT)
C8390C      INTEGRATE T AND Z EXPANSION FUNCTIONS OVER SLOC.
C8400C      COMPLEX TINT,ZINT
C8410C      COMPLEX U
C8415C      COMMON /BOT2/ NP,RL,YB(83),XB(83),Y91(82),XB1(82)
C8420C      DATA PI,U /3.14159265,(0.,1.)/
C8430C      IF(M.EQ.0) TINT=Z1-Z0
C8440C      IF(M.NE.0) TINT=-U*RL/M/PI*(CEXP(U*M*PI*Z1/BL)-
C8450C      1 CEXP(U*M*PI*Z0/BL))
C8460C      ZINT=TINT-(-1)**M*(Z1-Z0)
C8470C      RETURN
C8480C      END
C8490C -----
C8500C      SUBROUTINE PLOT(X,Y,ISYM,N,LX,LY,NC,NR)
C8510C -----
C8520C
C8530C      WRITTEN 2/14/74  BY J. M. PUTNAM      DEPT 220      X23877
C8540C
C8550C      THIS ROUTINE PRODUCES A LINEAR XY PLOT.
C8560C      N IS THE NUMBER OF POINTS TO BE PLOTTED.
C8570C      NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
C8580C      NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
C8590C      NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
C8600C
C8610C      DIMENSION X(1),Y(1),ISYM(1),HEAD(10)
C8620C      INTEGER LINE(102),BLANK,PLUS
C8630C      DATA BLANK,PLUS /1H,1H+/
C8640C      NC1=NC+1
C8650C      WRITE(6,501)
C8660 501  FORMAT(1H1)
C8670C      N10=(NC-1)/10
C8680C      XMIN=X(1)
C8690C      XMAX=X(1)
C8700C      YMIN=Y(1)
C8710C      YMAX=Y(1)
C8720C      DO 6 I=1,N
C8730C      IF(X(I).LT.XMIN) XMIN=X(I)
C8740C      IF(X(I).GT.XMAX) XMAX=X(I)
C8750C      IF(Y(I).LT.YMIN) YMIN=Y(I)
C8760C      IF(Y(I).GT.YMAX) YMAX=Y(I)

```

```

1677C 6 CONTINUE
1678C DELX=(XMAX-XMIN)/(NC-1)
1679C DELY=(YMAX-YMIN)/(NR-1)
1680C DO 5 I=1,N10
1681C 5 HEAD(I)=XMIN+10*I*DELX
1682C DO 11 K=1,NC1
1683C LINE(K)='- '
1684C IF(MOD(K,5).EQ.1) LINE(K)='-.'
1685C 11 CONTINUE
1686C WRITE(6,504) (LINE(K),K=1,NC1)
1687C YU=YMAX+DELY/2.
1688C YL=YMAX-DELY/2.
1689C DO 7 J=1,NR
1690C DO 8 K=1,NC
1691C 8 LINE(K)=BLANK
1692C LINE(NC1)='I'
1693C DO 9 I=1,N
1694C IF(Y(I).GE.YU) GO TO 9
1695C IF(Y(I).LT.YL) GO TO 9
1696C K=(X(I)-XMIN)/DELX+1.5
1697C IF(LINE(K).NE.BLANK) LINE(K)=PLUS
1698C IF(LINE(K).EQ.BLANK) LINE(K)=ISYM(I)
1699C 9 CONTINUE
1700C YN=(YU+YL)/2.
1701C WRITE(6,508) YN,(LINE(K),K=1,NC1)
1702C YU=YL
1703C YL=YL-DELY
1704C 7 CONTINUE
1705C DO 12 K=1,NC1
1706C LINE(K)='- '
1707C IF(MOD(K,5).EQ.1) LINE(K)='-.'
1708C 12 CCNTINUE
1709C WRITE(6,504) (LINE(K),K=1,NC1)
1710C DO 13 K=1,NC1
1711C LINE(K)=BLANK
1712C IF(MOD(K,10).EQ.1) LINE(K)='-I'
1713C 13 CONTINUE
1714C WRITE(6,3002) LY,LX,(LINE(K),K=1,NC1)
1715C WRITE(6,507) XMIN,(HEAD(I),I=1,N10)
1716C WRITE(6,502) N
1717C 5C2 FORMAT(/,2X,I4,15H POINTS PLOTTED)
1718C RETURN
1719C 5C4 FORMAT(1X,14(14-),102A1)
1720C 507 FORMAT(1CX,11(F10.4))
1721C 5C8 FORMAT(1X,F12.4,1X,1HI,102A1)
1722C 3C02 FORMAT(4X,A1,5H / ,A1,4X,102A1)
1723C END
1724C/ECR

```



```

C010C BOTR, P30, T1CC, CM15CC00, STCX3.
C0110 ACCOUNT, MC3265, BOT2A.
C0120 BANNERS(OUTPUT) *J. PUTNAM*DEPT 220*BLD 110-4**
C0130 ATTACH(INFIL=YSSX)
C0140 FTN(R=0, OPT=1)
C0164 LDSET(PRESET=INDEF)
C0170 CLGO.
C0190 /EOR
C0200      PROGRAM BOTR(INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT,
C0210      1 INFIL, TAPE1=INFIL)
C0220C
C0230C BOTR IS THE BOT RADIATION CODE.
C0240C
C0250C UNIT 5 IS THE CARD READER.
C0260C UNIT 6 IS THE LINE PRINTER.
C0270C UNIT 1 IS A DISK FILE CONTAINING THE Y MATRIX.
C0280C
C0285      COMPLEX U
C0290      COMPLEX VB(82), VW(59)
C0295      COMPLEX GT(73), GP(73)
C0300      COMPLEX CR(1680), CW(59), CC(96)
C0305      COMPLEX ESC(3), YSC(3)
C0310      COMPLEX Y(2304)
C0315      COMPLEX RBT(62), RBP(82), RWT(59), RWP(59), RCT(96), RCP(96)
C0320      COMMON /PLOT1/ NPLT, XPLT(200), YPLT(200), ZPLT(200), ISYM(200)
C0325      COMMON /WAVE/ RK
C0330      COMMON /BOT1/ NMODE, NPT, N9AND
C0335      COMMON /BOT2/ NP, BL, YB(83), XB(83), YB1(82), XB1(82)
C0340      COMMON /BOT3/ DH(82), SV(82), CV(82)
C0345      COMMON /BOT5/ T(160), TP(150), TZ(160)
C0350      COMMON /BOT6/ IEDGE, IUNIF
C0355      COMMON /SLOT1/ NSA, IS(20), ZO(20), Z1(20)
C0360      COMMON /WIRE1/ NPW, XW(101), YW(101), ZW(101),
C0365      1 XW1(100), YW1(100), ZW1(100)
C0370      COMMON /WIRE4/ LW, TW(196), TPW(196), INDTW(49)
C0375      COMMON /JUNC1/ NJ, INDJW(10), RADJ(10), RADD(10)
C0380      COMMON /CAP1/ NC, XC, YC, ZC(2)
C0385      COMMON /CAP2/ NPR, RHOC(21), RHOC1(20), DRHOC(20)
C0390      COMMON /EDG1/ NE, ZE(2), ZBE(10)
C0395      COMMON /INT/ M, RH02, ZP
C0400      DIMENSION THR(73), PHIP(73)
C0405      DIMENSION ANG(6), IPLANE(6), ANG1(6), ANG2(6)
C0410      DIMENSION HORD(2)
C0415      NPLT=0
C0420      U=(0., 1.)
C0425      ETA=376.707
C0430      PI=3.14159265
C0435      DTOR=PI/180.
C0440      READ(5, 1) BK
C0445      FORMAT(E15.7)
C0450      1 WRITE(6, 2) BK
C0455      2 FORMAT(9H1
C0460      2
C0465      2

```

```

RK, /, E15.7)

```

```

CC47CC
CC480      CALL BOTIN
CC490C
CC500C      CALL CAPIN
CC505      NCV=C
CC51CC
CC520C      CALL WIPEIN
CC522C
CC540C      CALL SLOTIN
CC550C
CC552      IF(NPW.NE.C) CALL VMIRE(NWJV,VW)
CC554C
CC560C READ INPUTS, DESCRIBING THE RADIATION PLANES.
CC570      READ(5,44) NANG,NT
CC580 44    FORMAT(2I3)
CC590      READ(5,53) (ANG(I),I=1,NANG)
CC600 53    FORMAT(10F8.4)
CC610C      IPLANE = 1 FOR PHI FIXED.
CC620C      2 FOR THETA FIXED.
CC630      READ(5,54) (IPLANE(I),I=1,NANG)
CC640 54    FORMAT(1CI8)
CC642      READ(5,53) (ANG1(I),I=1,NANG)
CC644      READ(5,53) (ANG2(I),I=1,NANG)
CC650      WRITE(6,3) NANG,NT
CC660 3    FORMAT(//,25H NUMBER OF FIXED ANGLES =,I3,//
CC670      1 35H NUMBER OF ANGLES PER FIXED ANGLE =,I4)
CC680      WRITE(6,4) (ANG(I),IPLANE(I),ANG1(I),ANG2(I),I=1,NANG)
CC690 4    FORMAT(//,45H FIXED ANGLE CODE VARIABLE ANGLE RANGE,
CC700      1 //,(F8.1,9X,I2,6X,F8.1,3H - ,F8.1))
CC710      LS=NP-3
CC712      NM=LS/2
CC714      NP1=NM+1
CC716      NM4=NM*4
CC720      LSS=LS*LS
CC722      LW=(NPR-3)/2
CC724      LC=NC+NM*LR
CC726      LC2=LC*2
CC728      LE=NE+NM
CC730      LSC=LS*(LC2+LE)
CC735      LWJ=LW+NJ
CC740      LSM=LS*LWJ
CC750      LCC=(LC2+LE)**2
CC760      LCW=(LC2+LE)*LWJ
CC770      LWW=LWJ**2
CC810      BKL=BK*BL
CC830      KMODE=2*NMODE-1
CC840      CALL ZERO(LS*KMODE,CR)
CC860      IF(NC.NE.0) CALL ZERO(LC2+LE,CC)
CC870      IF(NPW.NE.C) CALL ZERO(LWJ,CW)
CC880      CALL CORDER(MORD)
CC890C CALCULATE CURRENTS, EO. 10.
CC900      NUMR=0

```

```

0091C 5C CONTINUE
0092C READ(1) M,N
0093C IF(EOF(1)) 37,60
0094C 60 NUMB=NUMB+1
0095C IC=(M+NMCD-1)*LS+1
0096C DETERMINE SUBMATRIX TYPE, AND CALCULATE CURRENTS.
0097C IF(N.LT.NMODE) GO TO 61
0098C IF(N.EQ.MORD(1)) GO TO 62
0099C IF(N.EQ.MORD(2)) GO TO 63
0100C GO TO 50
0101C 61 IF(NSA.EQ.C) GO TO 80
0102C CALL VBOT(N,VR)
0103C IF(M.LT.NMCD) GO TO 71
0104C IF(M.EQ.MORD(1)) GO TO 72
0105C IF(M.EQ.MORD(2)) GO TO 73
0106C GO TO 50
0107C 62 IF(NCV.EQ.C) GO TO 80
0108C IF(M.LT.NMCD) GO TO 74
0109C IF(M.EQ.MORD(1)) GO TO 75
0110C IF(M.EQ.MORD(2)) GO TO 76
0111C GO TO 50
0112C 63 IF(NWJV.EQ.C) GO TO 80
0113C IF(M.LT.NMCD) GO TO 77
0114C IF(M.EQ.MORD(1)) GO TO 78
0115C IF(M.EQ.MORD(2)) GO TO 79
0116C GO TO 50
0117C BOT-BOT.
0118C 71 READ(1) (Y(I),I=1,LSS)
0119C CALL MULTYV(LS,LS,Y,VB,CB(IC))
0120C GO TO 50
0121C CAP-ROT.
0122C 72 READ(1) (Y(I),I=1,LSC)
0123C CALL MULTYV(LC2+LE,LS,Y,VR,CC)
0124C GO TO 50
0125C WIRE-BOT.
0126C 73 READ(1) (Y(I),I=1,LSW)
0127C CALL MULTYV(LWJ,LS,Y,VR,CW)
0128C GO TO 50
0129C BOT-CAP.
0130C 74 READ(1) (Y(I),I=1,LSC)
0131C CALL MULTYV(LS,LC2+LE,Y,VC,CB(IC))
0132C GO TO 50
0133C CAP-CAP.
0134C 75 READ(1) (Y(I),I=1,LCC)
0135C CALL MULTYV(LC2+LE,LC2+LE,Y,VC,CC)
0136C GO TO 50
0137C WIRE-CAP.
0138C 76 READ(1) (Y(I),I=1,LCW)
0139C CALL MULTYV(LWJ,LC2+LE,Y,VC,CW)
0140C GO TO 50
0141C BOT-WIRE.
0142C 77 READ(1) (Y(I),I=1,LSW)

```

```

01410      CALL MULTYV(LS,LWJ,Y,VW,CB(IC))
01420      GO TO 5C
01430C CAP-WIRE.
01440 78 READ(1) (Y(I),I=1,LCW)
01450      CALL MULTYV(LC2+LE,LWJ,Y,VW,CC)
01460      GO TO 5C
01470C WIRE-WIRE.
01480 79 READ(1) (Y(I),I=1,LWW)
01490      CALL MULTYV(LWJ,LWJ,Y,VW,CW)
01500      GO TO 5C
01542 80 READ(1) Y(1)
01544      GO TO 5C
01550C ALL Y SUBMATRICES HAVE BEEN READ.
01560 37 CONTINUE
01562      WRITE(6,114) NUMB
01564 114 FORMAT(/,1X,I4,17H SUBMATRICES READ)
01570C CALCULATE PC.
01580      PC=0.C
01581      IF(NSA.EQ.0) GO TO 85
01582      DO 82 MM=1,KMODE
01583      M=-NPCOE+MM
01584      IC=(M+NMODE-1)*LS+1
01585      CALL VBOT(M,VR)
01586      CALL MULTVC(LS,VB,CB(IC),PO)
01587 82 CONTINUE
01588 85 CONTINUE
01589      IF(LC.NE.0 .AND. NCV.NE.0) CALL MULTVC(LC2+LE,VC,CC,PO)
01590      IF(NPW.NE.0 .AND. NWJV.NE.0) CALL MULTVC(LWJ,VW,CW,PO)
01591      WRITE(6,121) D9CON(PO)
01600 121 FORMAT(/,18H TOTAL POWER(DB) =,F8.2)
01630C *** END CALCULATION OF CURRENTS ***
01640      P1=8K*8K*ETA/4./PI/PO
01650      DO 9C IANG=1,NANG
01660C INITIALIZE ARRAYS, AND SET THE RADIATION ANGLES.
01670      IPLAN=IPLANE(IANG)
01675      DT=(ANG2(IANG)-ANG1(IANG))/(NT-1)
01680      DO 10C K=1,NT
01690      GT(K)=0.0
01700      GP(K)=0.0
01710      PHIR(K)=ANG(IANG)*(2-IPLAN)+(ANG1(IANG)+DT*(K-1))*(IPLAN-1)
01720 10C THR(K)=ANG(IANG)*(IPLAN-1)+(ANG1(IANG)+DT*(K-1))*(2-IPLAN)
01730      WRITE(6,403)
01740 403 FORMAT(1H1,25X,9HPOWER(DB),/,20X,21(1H-),/,
01750      1 39H      PHI      THETA      0      0,/,
01760      2 39H      -      -      -      /,/)
01770C CALCULATION OF FAR-FIELD RADIATION.
01780      DO 11C K=1,NT
01790      CALL RBOT(RBT,RBP,1,THR(K),PHIR(K))
01800      IC=0
01810      DO 115 MM=1,KMODE
01820      M=-NPCOE+MM
01830C CALCULATE DM AS A FUNCTION OF THETA (EQ. 41).

```

```

C1840      OPT=2.*RL*SINC(4*PI+BKL*205(THR(K)*DTOR))
C1850      DMZ=DMT-2.*BL*(-1)*M*SINC(9KL*205(THR(K)*DTOR))
C1860      DO 1C5 J=1,LS
C1870      IC=IC+1
C1880      DM=OPT
C1890      IF(J.GT.NM) DM=DMZ
C1900      GT(K)=GT(K)+DM*RR(J)*CR(IC)
C1910 1C5 GP(K)=GP(K)+DM*RR(J)*CB(IC)
C1920 115 CONTINUE
C1921      IF(LC.EQ.0) GO TO 106
C1922      CALL RCAP(RCT,RCO,1,THR(K),PHIR(K))
C1923      CALL MULTRC(LC2+LE,RCT,CC,GT(K))
C1924      CALL MULTRC(LC2+LE,RCO,CC,GP(K))
C1925 1C6 CONTINUE
C1926      IF(NPW.EC.C) GO TO 107
C1927      CALL RWIRE(RWT,RWP,1,THR(K),PHIR(K))
C1928      CALL MULTRC(LWJ,RWT,CW,GT(K))
C1929      CALL MULTRC(LWJ,RWP,CW,GP(K))
C1930 1C7 CONTINUE
C1931      X2=CABS(GT(K))
C1940      X2=DBCON(P1*X2*X2)
C1950      X3=CABS(GP(K))
C1960      X3=DBCON(P1*X3*X3)
C1970      WRITE(6,113) PHIR(K),THR(K),X2,X3
C1980 113 FORMAT(1X,2F8.1,2F12.2)
C1990 11C CONTINUE
C2000 9C CONTINUE
C2010C *** END FAP-FIELD RADIATION ***
C2020C LIST AND PLOT CURRENTS.
C2030C CALL LPCUR(NMODE,NM,CB,LW,NJ,CW,LC,LE,CC)
C2060C **** END LISTING AND PLOTTING OF CURRENTS ****
C2080C
C2090C NEAR FIELD ANALYSIS FOLLOWS.
C2100C
C2110C NOTE, ESC IS FOR ELECTRIC FIELDS
C2120C      HSC IS FOR MAGNETIC FIELDS
C2130C      INDICES 1,2,AND 3 ARE FOR THE X,Y,AND Z COMPONENTS OF THE
C2140C      FIELDS, RESPECTIVELY.
C2150C
C2160C      READ(5,44) NTEST
C2170C      IF(NTEST.EQ.0) GO TO 900
C2180C      WRITE(6,404)
C2190C 4C4 FORMAT(1H1,/,214 NEAR FIELD ANALYSIS,/)
C2200C      1 59X,7HE-FIELD,37X,7HH-FIELD,/,
C2210C      2 43X,39(1H-),5X,39(1H-),/,
C2220C      3 55X,16HFIELD COMPONENTS,28X,16HFIELD COMPONENTS,/,
C2230C      4 24H 7TEST YTEST XTEST
C2240C      5 11X,2(14X,6H X 7X,5H Y ,9X,34 Z ),/)
C2250C      DO 49C ITEST=1,NTEST
C2260C      READ(5,53) ZIFST,YTEST,XTEST
C2262C UPDATE PLCT ARRAY.
C2264C      NPLOT=NPLOT+1

```

```

C2266      XPL0T(NPLOT)=XTEST
C2268      YPLOT(NPLOT)=YTEST
C2270      ZPLOT(NPLOT)=ZTEST
C2272      ISYM(NPLOT)='N'
C2330      ESC(1)=0.
C2340      ESC(2)=C.
C2350      ESC(3)=C.
C2360      HSC(1)=0.
C2370      HSC(2)=C.
C2380      HSC(3)=C.
C2385C BOT CONTRIBUTION.
C2390C STEP THROUGH MODES.
C2400      M=-NM0DE
C2410      DO 43C MM=1,KM0DE
C2420      M=M+1
C2430C CALCULATE M-TH MODAL CURRENT COEFFICIENTS (EQS. 67 & 71).
C2440      CALL NEARB(XTEST,YTEST,ZTEST,Y)
C2450C CALCULATE AND SUM Y*CB USING THE FIRST 6 ROWS OF Y, YIELDING THE
C2460C 3 FIELD COMPONENTS FOR BOTH ELECTRIC AND MAGNETIC NEAR FIELDS,
C2470C (EQS. 67 & 71).
C2480      DO 42C K=1,3
C2490      I1=(M+NM0DE-1)*LS
C2500      J1=K-6
C2510      DO 42C I=1,LS
C2520      I1=I1+1
C2530      J1=J1+6
C2540      J2=J1+3
C2545      ESC(K)=ESC(K)+Y(J1)*CB(I1)
C2550      HSC(K)=HSC(K)+Y(J2)*CB(I1)
C2551 42C CONTINUE
C2552 43C CONTINUE
C2554      IF(INC.EQ.0) GO TO 450
C2555C CAP CONTRIBUTION.
C2556      CALL NEARC(XTEST,YTEST,ZTEST,Y)
C2557      DO 44C K=1,3
C2558      J1=K-6
C2559      DO 44C I=1,LC2
C2560      J1=J1+6
C2561      J2=J1+3
C2562      ESC(K)=ESC(K)+Y(J1)*CC(I)
C2563      HSC(K)=HSC(K)+Y(J2)*CC(I)
C2564 44C CONTINUE
C2565 450 IF(NPW.EQ.C) GO TO 47C
C2566C WIRE CONTRIBUTION.
C2567      CALL NEARW(XTEST,YTEST,ZTEST,Y)
C2568      DO 46C K=1,3
C2569      J1=K-6
C2570      DO 46C I=1,LW
C2571      J1=J1+6
C2572      J2=J1+3
C2573      ESC(K)=ESC(K)+Y(J1)*CW(I)
C2574      HSC(K)=HSC(K)+Y(J2)*CW(I)

```

```

C2575 460 CONTINUE
C2590 470 WRITE(6,505) ZTEST,YTEST,XTEST
C2600 505 FORMAT(7,1X,3F8.4)
C2780 WRITE(6,506) (CABS(ESC(I)),I=1,3),(CABS(HSC(I)),I=1,3)
C2790 506 FORMAT(43X,3E13.4,5X,3E13.4)
C2800 490 CONTINUE
C2810 900 CONTINUE
C2812 CALL PLOT(XPLOT,YPLOT,ISYM,NPLOT,'X','Y',101,51)
C2814 CALL PLOT(ZPLOT,YPLOT,ISYM,NPLOT,'Z','Y',101,51)
C2816 CALL PLOT(ZPLOT,XPLOT,ISYM,NPLOT,'Z','X',101,51)
C2818 STOP
C2820 END
C2821C -----
C2822 SUBROUTINE ZEP7(N,A)
C2823C A=C.
C2824 COMPLEX A(1)
C2825 DO 100 I=1,N
C2826 A(I)=C.C
C2827 100 CONTINUE
C2828 RETURN
C2829 END
C2830C -----
C2831C SUBROUTINE MULTYV(NI,NJ,Y,V,C)
C2832C UPDATE CURRENT ARRAY C BY Y * V.
C2833C
C2834C COMPLEX S,Y(1),V(1),C(1)
C2835 DO 100 I=1,NI
C2836 S=0.C
C2837 IND=I
C2838 DO 50 J=1,NJ
C2839 S=S+Y(IND)*V(J)
C2840 50 IND=IND+NI
C2841 C(I)=C(I)+S
C2842 100 CONTINUE
C2843 RETURN
C2844 END
C2845C -----
C2846C SUBROUTINE MULTVC(NI,V,C,P)
C2847C UPDATE P BY V(CONJUGATE) * C.
C2848C
C2849C COMPLEX V(1),C(1)
C2850 DO 100 I=1,NI
C2851 P=P+C*CONJG(V(I))*C(I)
C2852 100 CONTINUE
C2853 RETURN
C2854 END
C2855C -----
C2856C SUBROUTINE MULTRC(NI,P,C,G)
C2857C UPDATE G BY R * C.

```

```

C3140C
C3150C COMPLEX G,R(1),C(1)
C3160C DO 100 I=1,NI
C3170C G=G+R(1)*C(I)
C3180C 100 CONTINUE
C3190C RETURN
C3200C END
C3210C *****
C3220C FUNCTION SINC(X)
C3230C *****
C3240C SINC=1./X
C3250C IF(X.NE.C.C) SINC=SIN(X)/X
C3260C RETURN
C3270C END
C3280C *****
C3290C FUNCTION DRCON(X)
C3300C *****
C3310C RETURN X IN DA.
C3320C DBCON=-1000.C
C3330C IF(X.EQ.0.0) RETURN
C3340C DBCON=1C.*ALOG10(X)
C3350C RETURN
C3360C END
C3370C -----
C3375C SUBROUTINE ROTIN
C3380C
C3385C READ POT COORDINATES AND COMPUTE ROT SEGMENT ARRAYS.
C3390C
C3392C COMMON /PLOT1/ NPLOT,XPLOT(200),YPLLOT(200),ZPLOT(200),ISYM(200)
C3395C COMMON /BOT1/ NMODE,NPT,NBAND
C3400C COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
C3405C COMMON /BOT3/ DM(82),SV(82),CV(82)
C3410C COMMON /BOT5/ T(160),TP(160),TZ(160)
C3415C COMMON /BOT6/ IEDGE,IUNIF
C3420C READ(5,49) NMODE,NPT,NBAND
C3425C READ(5,49) NP
C3430C 49 FORMAT(3I3)
C3435C WRITE(6,48) NMODE,NPT,NBAND,NP
C3440C 48 FORMAT(32H NMODE NPT NBAND NP,/,4I8)
C3445C READ(5,53)(YB(I),I=1,NP)
C3450C READ(5,53)(XB(I),I=1,NP)
C3455C 53 FORMAT(10F8.4)
C3460C WRITE(6,55)
C3465C 55 FORMAT(/,3H YB)
C3470C WRITE(6,46)(YB(I),I=1,NP)
C3475C 46 FORMAT(1X,10F8.4)
C3480C WRITE(6,56)
C3485C 56 FORMAT(/,3H XB)
C3488C WRITE(6,46)(XB(I),I=1,NP)
C3490C PLOT THE BODY COORDINATES.
C3492C CALL PLOT8(XB,YB,NP,41)
C3492C READ(5,53) BL

```

```

C3494      WRITE(6,47) RL
C3496 47    FORMAT(/,21H HALF-LENGTH OF BOT =,F12.4)
C3500C    UPDATE PLOT ARRAY.
C3501      DO 15C I=1,NP
C3502        NPLOT=NPLOT+1
C3503        XPLOT(NPLOT)=X9(I)
C3504        YPLOT(NPLOT)=Y8(I)
C3505        ZPLOT(NPLOT)=BL
C3506        ISYM(NPLOT)='R'
C3507        NPLOT=NPLOT+1
C3508        XPLOT(NPLOT)=X9(I)
C3509        YPLOT(NPLOT)=Y8(I)
C3510        ZPLOT(NPLOT)=-BL
C3511        ISYM(NPLOT)='R'
C3512 15C  CONTINUE
C3520C    CHECK IF BOT CONTAINS AN ODD NUMBER OF POINTS.
C3522      IF(MOD(NP,2).NE.1) GO TO 980
C3530C    CHECK IF GENERATING CURVE IS OPEN OR CLOSED.
C3535      IEDGE=1
C3540      IF((Y8(1)-Y8(NP)).NE.0..OR.(X8(1)-X8(NP)).NE.0.) GO TO 10
C3545      IEDGE=0
C3550      Y8(NP+1)=Y8(2)
C3555      X8(NP+1)=X8(2)
C3560      Y8(NP+2)=Y8(3)
C3565      X8(NP+2)=X8(3)
C3570      NP=NP+2
C3575      WRITE(6,66) NP
C3580 66    FORMAT(/,39H BOT GENERATING CURVE IS CLOSED. NP = ,I3)
C3585 10    CONTINUE
C3590C    COMPUTATION OF BODY SEGMENT PARAMETERS.
C3595      DO 57 I=2,NP
C3600        I2=I-1
C3605        RR1=Y8(I)-Y8(I2)
C3610        RR2=X8(I)-X8(I2)
C3615        DH(I2)=SQRT(RR1*RR1+RR2*RR2)
C3620        XB1(I2)=.5*(X8(I)+X8(I2))
C3625        YB1(I2)=.5*(Y8(I)+Y8(I2))
C3630        SV(I2)=RR1/DH(I2)
C3635        CV(I2)=RR2/DH(I2)
C3640 57    CONTINUE
C3645C    CHECK IF BOT SEGMENTATION IS UNIFORM.
C3650      IUNIF=0
C3655      NP1=NP-1
C3660      DO 6C I=2,NP1
C3665        RR1=DH(I)/DH(1)
C3670        IF(RR1.LT.0.99..OR. RR1.GT.1.01) GO TO 2C
C3675 6C    CONTINUE
C3680      IUNIF=1
C3685      WRITE(6,67)
C3690 67    FORMAT(/,1 BOT GENERATING CURVE HAS UNIFORM SEGMENTATION')
C3695 20    CONTINUE
C3700C    COMPUTATION OF TRIANGLE FUNCTIONS T.

```

```

C3705      NM=(NP-3)/2
C3710      DO 74 J=1,NM
C3715      J2=2*(J-1)+1
C3720      J3=J2+1
C3725      J4=J3+1
C3730      J5=J4+1
C3735      J6=4*(J-1)+1
C3740      J7=J6+1
C3745      J8=J7+1
C3750      J9=J8+1
C3755      DEL1=DH(J2)+DH(J3)
C3760      DEL2=DH(J4)+DH(J5)
C3765      TP(J6)=1./DEL1
C3770      TP(J7)=1./DEL1
C3775      TP(J8)=-1./DEL2
C3780      TP(J9)=-1./DEL2
C3785      T(J6)=DH(J2)/2./DEL1
C3790      T(J7)=(DH(J2)+DH(J3)/2.)/DEL1
C3795      T(J8)=(DH(J4)/2.+DH(J5))/DEL2
C3800      T(J9)=(DH(J5)/2.)/DEL2
C3805      74 CONTINUE
C3810      NM4=NP+4
C3815      DO 75 J=1,NM4
C3820      75 T2(J)=T(J)
C3825      IF(IEQGEQ,0) GO TO 76
C3830      T2(1)=2.-T(1)
C3835      T2(2)=2.-T(2)
C3840      T2(NM4-1)=2.-T(NM4-1)
C3845      T2(NM4)=2.-T(NM4)
C3850      76 CONTINUE
C3855      RETURN
C3860      980 WRITE(6,981)
C3865      981 FORMAT(/, ' **** ERROR IN ROT INPUT ')
C3870      STOP
C3875      END
C3880C *****
C3882C SUBROUTINE PLOT9(X,Y,N,NR)
C3884C *****
C3886C
C3888C WRITTEN 2/14/74 BY J. M. PUTNAM DEPT 22G X23877
C3890C
C3892C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
C3894C N IS THE NUMBER OF POINTS TO BE PLOTTED.
C3896C NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
C3898C NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
C3900C NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
C3902C
C3904C REAL X(1),Y(1),HEAD(10)
C3906C INTEGER LINE(101),BLANK,STAR,PLUS
C3908C DATA BLANK,STAR,PLUS /14,14*,14*/
C3910C NC=51
C3912C N10=(NC-1)/10

```

```

C3914      WRITE(6,500)
C3916 500  FORMAT(//,44H BODY COORDINATES, + INDICATES TRIANGLE PEAK)
C3918      WRITE(6,504)
C3920      XMIN=X(1)
C3922      XMAX=X(1)
C3924      YMIN=Y(1)
C3926      YMAX=Y(1)
C3928      DO 6 I=1,N
C3930      IF(X(I).LT.XMIN) XMIN=X(I)
C3932      IF(X(I).GT.XMAX) XMAX=X(I)
C3934      IF(Y(I).LT.YMIN) YMIN=Y(I)
C3936      IF(Y(I).GT.YMAX) YMAX=Y(I)
C3938 6      CONTINUE
C3940      DEL=XMAX-XMIN
C3942      IF(YMAX-YMIN.GT.DEL) DEL=YMAX-YMIN
C3944      XMAX=XMIN+DEL
C3946      YMAX=YMIN+DEL
C3948      DO 5 I=1,N10
C3950      Z=I
C3952 5      HEAD(I)=(XMAX-XMIN)*Z/N10+XMIN
C3954      DY=(YMAX-YMIN)/(N0-1)
C3956      IEDGE=1
C3958      IF(X(1).EQ.X(N) .AND. Y(1).EQ.Y(N)) IEDGE=0
C3960      Z=YMAX+DY
C3962      YL=Z-DY/2.
C3964      DO 7 J=1,NR
C3966      DO 8 K=1,NC
C3968 8      LINE(K)=BLANK
C3970      Z=Z-DY
C3972      YU=YL
C3974      YL=Z-DY/2.
C3976      DO 9 I=1,N
C3978      IF(Y(I).GE.YU) GO TO 9
C3980      IF(Y(I).LT.YL) GO TO 9
C3982      K=(X(I)-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
C3984      IF(K.GT.NC) K=NC
C3986      LINE(K)=STAR
C3988      IF(MOD(I,2).EQ.1) LINE(K)=PLUS
C3990      IF(IEEDGE.EQ.0) GO TO 9
C3992      IF(I.EQ.1 .OR. I.EQ.N) LINE(K)=STAP
C3994 9      CONTINUE
C3996      WRITE(6,508) Z,(LINE(K),K=1,NC)
C3998 7      CONTINUE
C4000      WRITE(6,504)
C4002      WRITE(6,3002)
C4004      WRITE(6,507) XMIN,(HEAD(I),I=1,N10)
C4006      RETURN
C4008 504  FORMAT ( 1X, 14(14-), 14., 10(54----), 14- )
C4010 507  FORMAT(10X,11(F10.4))
C4012 508  FORMAT (1X, F12.4,1X, 14I, 51A1, 14I )
C4014 3002  FORMAT(4X,7HYH / XH,4X,1HI,5(9X,1HI))
C4016      END

```

```

C4410C -----
C4420C SUBROUTINE CAPIN
C4430C
C4440C READ CAP INPUTS AND COMPUTE CAP ARRAYS.
C4450C
C4460C COMMON /PLOT1/ NPLCT,XPLOT(200),YPLT(200),ZPLOT(200),TSYM(200)
C4470C COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
C4480C COMMON /CAP1/ NC,XC,YC,ZC(2)
C4490C COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
C4500C COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
C4510C COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
C4520C COMMON /EDG1/ NE,ZE(2),TRE(10)
C4530C COMMON /EDG2/ TCE(10),TPCE(10),TBE(10),TPBE(10)
C4540C READ(5,1) NC,NPR,NF
C4550C 1 FORMAT(3I3)
C4560C IF(NE.NE.0) NE=NC
C4570C WRITE(6,3) NC,NPR,NE
C4580C 3 FORMAT(16H1 NC NPR NE,/,3I5,/)
C4590C IF(NC.EQ.0) RETURN
C4600C READ(5,2) XC,YC
C4610C 2 FORMAT(10F8.4)
C4620C READ(5,2) (ZC(I),I=1,NC)
C4630C READ(5,2) (RHOC(I),I=1,NPR)
C4640C IF(NE.NE.0) READ(5,2) (ZE(I),I=1,NE)
C5100C UPDATE PLOT ARRAY.
C5102C DO 150 I=1,NC
C5104C NPLCT=NPLCT+1
C5106C XPLOT(NPLCT)=XC
C5108C YPLOT(NPLCT)=YC
C5110C ZPLOT(NPLCT)=ZC(I)
C5112C ISYM(NPLCT)='C'
C5114C IF(NE.EQ.0) GO TO 150
C5116C NPLCT=NPLCT+1
C5118C XPLOT(NPLCT)=XB(1)
C5120C YPLOT(NPLCT)=YB(1)
C5122C ZPLOT(NPLCT)=ZE(I)
C5124C ISYM(NPLCT)='E'
C5126C 150 CONTINUE
C5144C WRITE(6,4)
C5146C 4 FORMAT(37H CAP XC YC ZC ZE,/)
C5148C DO 100 I=1,NC
C5149C IF(NE.EQ.0) WRITE(6,5) I,XC,YC,ZC(I)
C5150C IF(NE.NE.0) WRITE(6,5) I,XC,YC,ZC(I),ZE(I)
C5151C 100 CONTINUE
C5152C 5 FORMAT(14,4X,4F8.4)
C5154C WRITE(6,6)
C5156C 6 FORMAT(7,5H RHOC)
C5158C WRITE(6,7) (RHOC(I),I=1,NPR)
C5160C 7 FORMAT(1X,10F8.4)
C5162C IF(MOD(NPR,2).NE.1) GO TO 980
C5164C DO 120 I=2,NPR
C5166C 120 IF(RHOC(I).LE.RHOC(I-1)) GO TO 980

```

```

05168C COMPUTATION OF CAP SECTOR PARAMETERS.
05170 DO 47 I=1,NP
05172 RC(I)=SQRT((YR(I)-YC)**2+(XR(I)-XC)**2)
05174 47 CONTINUE
05176 DO 57 I=2,NP
05178 I2=I-1
05180 RR1=YB1(I2)-YC
05182 RR2=XB1(I2)-XC
05184 RC1(I2)=SQRT(RR1*RR1+RR2*RR2)
05186 AC(I2)=ABS((XR(I)-XC)*(YB(I)+YC)+(XB(I2)-XB(I))*
05188 1 (YB(I2)+YB(I))+(XC-XR(I2))*(YC+YB(I2)))/2.
05190 SPC(I2)=RR1/RC1(I2)
05192 CPC(I2)=RR2/RC1(I2)
05194 57 CONTINUE
05196 DO 67 I=2,NPR
05198 I2=I-1
05200 RHOC1(I2)=(RHOC(I)+RHOC(I2))/2.
05202 DRHOC(I2)=RHOC(I)-RHOC(I2)
05204 67 CONTINUE
05206C COMPUTATION OF CAP TRIANGLE FUNCTIONS.
05208 LC=(NPR-3)/2
05210 DO 74 J=1,LC
05212 J2=2*(J-1)+1
05214 J3=J2+1
05216 J4=J3+1
05218 J5=J4+1
05220 J6=4*(J-1)+1
05222 J7=J6+1
05224 J8=J7+1
05226 J9=J8+1
05228 DEL1=DRHOC(J2)+DRHOC(J3)
05230 DEL2=DRHOC(J4)+DRHOC(J5)
05232 TPCR(J6)=1./DEL1
05234 TPCR(J7)=1./DEL1
05236 TPCR(J8)=-1./DEL2
05238 TPCR(J9)=-1./DEL2
05240 TCR(J6)=DRHOC(J2)/2./DEL1
05242 TCR(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
05244 TCR(J8)=(DRHOC(J4)/2.+DRHOC(J5))/DEL2
05246 TCR(J9)=DRHOC(J5)/2./DEL2
05248 74 CONTINUE
05250 LC4=LC+4
05252 DO 75 I=1,LC4
05254 TCT(I)=TCR(I)
05256 TPCT(I)=TPCR(I)
05258 75 CONTINUE
05260 TCT(LC4-1)=2.-TCT(LC4-1)
05262 TCT(LC4)=2.-TCT(LC4)
05264 TPCT(LC4-1)=-TPCT(LC4-1)
05266 TPCT(LC4)=-TPCT(LC4)
05268 IF(RHOC(1).EQ.0.0) GO TO 76
05270 TCT(1)=2.-TCT(1)

```

```

05272      TCT(2)=2.-TCT(2)
05273      TPCT(1)=-TPCT(1)
05274      TPCT(2)=-TPCT(2)
05275      76      CONTINUE
05276      IF(INE.EQ.0) RETURN
05277      COMPUTATION OF EDGE HALF TRIANGLE FUNCTIONS.
05278      DO 80 IC=1,NC
05279      J2=NPR-2
05280      J3=J2+1
05281      J6=2*(IC-1)+1
05282      J7=J6+1
05283      DEL1=DRHOC(J2)+DRHOC(J3)
05284      TPCE(J6)=1./DEL1
05285      TPCE(J7)=1./DEL1
05286      TCE(J6)=DRHOC(J2)/2./DEL1
05287      TCE(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
05288      DEL2=ZC(IC)-ZC(IC)
05289      ZBE(J6)=ZC(IC)+0.25*DEL2
05290      ZBE(J7)=ZC(IC)+0.75*DEL2
05291      IF(DEL2.LT.0.0) GO TO 78
05292      10C      EDGE      IS AT Z=-L.
05293      TPBE(J6)=1./DEL2
05294      TPBE(J7)=1./DEL2
05295      TBE(J6)=0.75
05296      TBE(J7)=0.25
05297      GO TO 80
05298      78      CONTINUE
05299      10C      EDGE      IS AT Z=+L.
05300      TPBE(J6)=1./DEL2
05301      TPBE(J7)=1./DEL2
05302      TBE(J6)=-0.75
05303      TBE(J7)=-0.25
05304      80      CONTINUE
05305      RETURN
05306      980      WRITE(6,981)
05307      981      FORMAT(/, ' *** ERROR IN CAP INPUT')
05308      STOP
05309      END
05310      -----
05311      SUBROUTINE WIREIN
05312      READ WIRE COORDINATES AND COMPUTE WIRE SEGMENT ARRAYS.
05313      COMMON /PLOT1/ NPLOT,XPLOT(200),YPLOT(200),ZPLOT(200),ISYM(200)
05314      COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
05315      1 XW1(100),YW1(100),ZW1(100)
05316      COMMON /WIRE2/ OHW(100),DXW(100),DYW(100),DZW(100)
05317      COMMON /WIRE3/ NW,INDW(6),RADW(5)
05318      COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
05319      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
05320      COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
05321      COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)

```

```

C5575      COMMON /JUNC4/ UXJ(10),UYJ(10),UZJ(10)
C5580      COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
C5585      COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
C5590      NW=0
C5595      NJ=0
C5600      LW=0
C5605      READ(5,51) NW,NPW,NJ
C5610 51    FORMAT(3I3)
C5615      WRITE(6,1) NW,NPW,NJ
C5620 1     FORMAT(24H1      NW      NPW      NJ,/,3I8)
C5625      IF(NW.EQ.0) RETURN
C5630      READ(5,53) (XW(I),I=1,NPW)
C5635      READ(5,53) (YW(I),I=1,NPW)
C5640      READ(5,53) (ZW(I),I=1,NPW)
C5645 53    FORMAT(1CF8.4)
C5650      READ(5,52) (INDW(I),I=1,NW)
C5655 52    FORMAT(1CI8)
C5660      INDW(NW+1)=NPW+1
C5665      READ(5,53) (RADW(I),I=1,NW)
C5670      IF(NJ.EQ.0) GO TO 50
C5675C NOTE, INDJW MUST BE MONOTONIC INCREASING.
C5680      READ(5,52) (INDJW(I),I=1,NJ)
C5685      READ(5,53) (RADO(I),I=1,NJ)
C5690      READ(5,53) (UXJ(I),I=1,NJ)
C5695      READ(5,53) (UYJ(I),I=1,NJ)
C5700      READ(5,53) (UZJ(I),I=1,NJ)
C5705 5C    IERW=0
C5710      IERJ=0
C5715      WRITE(6,61)
C5720 61    FORMAT(/,22X,'WIRE COORDINATES',20X,'JUNCTION PARAMETERS',/,
C5725      1 26X,'IW',4X,'XW',6X,'YW',6X,'ZW',
C5730      2 10X,'IJ',4X,'RADO',5X,'UXJ',5X,'UYJ',5X,'UZJ')
C5735      IJ=1
C5740C THIS LOOP LISTS WIRE/JUNCTION POINTS, WHILE CHECKING THE FOLLOWING:
C5745C 1) EACH WIRE MUST CONTAIN AN ODD NUMBER OF POINTS.
C5750C 2) EACH JUNCTION MUST EITHER START OR TERMINATE A WIRE.
C5755C 3) CHECK THAT ALL JUNCTION POINTS ARE FOUND.
C5760      DO 100 IW=1,NW
C5765      WRITE(6,62) IW,RADW(IW)
C5770 62    FORMAT(2X,'WIRE',I3,' RADW=',F8.4)
C5775      I1=INDW(IW)
C5780      I2=INDW(IW+1)-1
C5785C CHECK FOR AN ODD NUMBER OF POINTS ON WIRE IW.
C5790      IF(MOD(I2-I1+1,2).NE.1) IERW=1
C5795      DO 9C I=I1,I2
C5800      WRITE(6,63) I,XW(I),YW(I),ZW(I)
C5805 63    FORMAT(25X,I3,3F8.4)
C5810C UPDATE PLOT ARRAY.
C5815      NPLDT=NPLDT+1
C5820      XPLOT(NPLDT)=XW(I)
C5825      YPLOT(NPLDT)=YW(I)
C5830      ZPLOT(NPLDT)=ZW(I)

```

```

05811      ISYM(NPLOT)='W'
05814C CHECK IF WIRE POINT I IS A JUNCTION POINT.
05815      IF(IJ.GT.NJ) GO TO 90
05820      IF(INDJW(IJ).NE.I) GO TO 90
05825C CHECK THAT JUNCTION POINT IJ IS AT THE START OR END OF WIRE.
05830      IF(I.NE.I1.AND. I.NE.I2) IERJ=1
05835      WRITE(6,64) IJ,RADD(IJ),UKJ(IJ),UYJ(IJ),UZJ(IJ)
05840 64  FORMAT('I',58X,I3,4F8.4)
05842C UPDATE PLOT ARRAY.
05843      ISYM(NPLOT)='J'
05845C DETERMINE DIRECTION IN WHICH WIRE LEAVES JUNCTION POINT.
05850      INDTJ(IJ)=1
05855      IF(I.EQ.I2) INDTJ(IJ)=-1
05860C COMPUTE JUNCTION PARAMETERS.
05865      RADJ(IJ)=RADW(IW)
05870      XJ(IJ)=XW(I)
05875      YJ(IJ)=YW(I)
05880      ZJ(IJ)=ZW(I)
05885      IJ=IJ+1
05890 90  CONTINUE
05895 100 CONTINUE
05900C CHECK FOR WIRE OR JUNCTION INPUT ERRORS.
05905      IF(IERW.NE.0) GO TO 980
05910      IF(IJ-1.NE.NJ.OR. IERJ.NE.0) GO TO 990
05915C COMPUTATION OF WIRE SEGMENT PARAMETERS.
05920      DO 57 I=2,NPW
05925      I2=I-1
05930      DXW(I2)=XW(I)-XW(I2)
05935      DYW(I2)=YW(I)-YW(I2)
05940      DZW(I2)=ZW(I)-ZW(I2)
05945      DHW(I2)=SQRT(DXW(I2)**2+DYW(I2)**2+DZW(I2)**2)
05950      XW1(I2)=0.5*(XW(I)+XW(I2))
05955      YW1(I2)=0.5*(YW(I)+YW(I2))
05960      ZW1(I2)=0.5*(ZW(I)+ZW(I2))
05965 57  CONTINUE
05970C COMPUTATION OF WIRE TRIANGLE FUNCTIONS TW.
05975      LW=0
05980      DO 75 IW=1,NW
05985      I1=INDW(IW)
05990      I2=INDW(IW+1)-1
05995      LW1=(I2-I1-2)/2
06000 75  DO 74 J=1,LW1
06005      LW=LW+1
06010      J2=2*(J-1)+I1
06015      J3=J2+1
06020      J4=J3+1
06025      J5=J4+1
06030      J6=4*(LW-1)+1
06035      J7=J6+1
06040      J8=J7+1
06045      J9=J8+1
06050      INDTW(LW)=J2

```

```

C6055      DEL1=DHW(J2)+DHW(J3)
C6060      DEL2=DHW(J4)+DHW(J5)
C6065      TPW(J6)=1./DEL1
C6070      TPW(J7)=1./DEL1
C6075      TPW(J8)=-1./DEL2
C6080      TPW(J9)=-1./DEL2
C6085      TW(J6)=DHW(J2)/2./DEL1
C6090      TW(J7)=(DHW(J2)+DHW(J3)/2.)/DEL1
C6095      TW(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
C6100      TW(J9)=DHW(J5)/2./DEL2
C6105  74  CONTINUE
C6110  75  CONTINUE
C6115C NOTE, LW=(NPW-NW)/2-NW
C6120      IF(INJ.EQ.0) RETURN
C6125C COMPUTATION OF JUNCTION HALF TRIANGLE FUNCTIONS TJ.
C6130      DO 85 IJ=1,NJ
C6135      IF(INDJ(IJ).GT.C) GO TO 80
C6140C JUNCTION IS AT THE END OF A WIRE.
C6145      J2=INDJW(IJ)-2
C6150      J3=J2+1
C6155      J6=2+(IJ-1)+1
C6160      J7=J6+1
C6165      INDJ(IJ)=J2
C6170      DEL1=-DHW(J2)-DHW(J3)
C6175      TPJ(J6)=1./DEL1
C6180      TPJ(J7)=1./DEL1
C6185      TJ(J6)=DHW(J2)/2./DEL1
C6190      TJ(J7)=(DHW(J2)+DHW(J3)/2.)/DEL1
C6195      GO TO 85
C6200C JUNCTION IS AT THE START OF A WIRE.
C6205  80  J4=INDJW(IJ)
C6210      J5=J4+1
C6215      J8=2+(IJ-1)+1
C6220      J9=J8+1
C6225      INDJ(IJ)=J4
C6230      DEL2=DHW(J4)+DHW(J5)
C6235      TPJ(J8)=-1./DEL2
C6240      TPJ(J9)=-1./DEL2
C6245      TJ(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
C6250      TJ(J9)=DHW(J5)/2./DEL2
C6255  85  CONTINUE
C6260C COMPUTE UNIT VECTORS FOR JUNCTION(DISK).
C6265      DO 95 IJ=1,NJ
C6270C NORMAL UNIT VECTOR (UJ).
C6275      RR=SGRT(UXJ(IJ)**2+UYJ(IJ)**2+UZJ(IJ)**2)
C6280      UXJ(IJ)=UXJ(IJ)/RR
C6285      UYJ(IJ)=UYJ(IJ)/RR
C6290      UZJ(IJ)=UZJ(IJ)/RR
C6295C FIND 2 ORTHOGONAL UNIT VECTORS IN THE PLANE OF THE DISK (UJ1 & UJ2).
C6300      UXJ1(IJ)=0.0
C6305      UYJ1(IJ)=0.0
C6310      UZJ1(IJ)=0.0

```

```

06315      UXJ2(IJ)=0.0
06320      UYJ2(IJ)=0.0
06325      UZJ2(IJ)=0.0
06330      IF(UXJ(IJ).EQ.0.0) GO TO 91
06335C FIND INTERSECTION WITH X-Y PLANE.
06340      UYJ1(IJ)=1.0
06345      UXJ1(IJ)=-UYJ(IJ)/UXJ(IJ)
06350C FIND INTERSECTION WITH X-Z PLANE.
06355      UZJ2(IJ)=1.0
06360      UXJ2(IJ)=-UZJ(IJ)/UXJ(IJ)
06365      GO TO 94
06370 91 IF(UYJ(IJ).EQ.0.0) GO TO 92
06375C FIND INTERSECTION WITH Y-Z PLANE.
06380      UZJ1(IJ)=1.0
06385      UYJ1(IJ)=-UZJ(IJ)/UYJ(IJ)
06390C FIND INTERSECTION WITH X-Y PLANE.
06395      UXJ2(IJ)=1.0
06400      UYJ2(IJ)=-UXJ(IJ)/UYJ(IJ)
06405      GO TO 94
06410 92 IF(UZJ(IJ).EQ.0.0) GO TO 94
06415C FIND INTERSECTION WITH X-Z PLANE.
06420      UXJ1(IJ)=1.0
06425      UZJ1(IJ)=-UXJ(IJ)/UZJ(IJ)
06430C FIND INTERSECTION WITH Y-Z PLANE.
06435      UYJ2(IJ)=1.0
06440      UZJ2(IJ)=-UYJ(IJ)/UZJ(IJ)
06445 94 CONTINUE
06450      RR=SQRT(UXJ1(IJ)**2+UYJ1(IJ)**2+UZJ1(IJ)**2)
06455      UXJ1(IJ)=UXJ1(IJ)/RR
06460      UYJ1(IJ)=UYJ1(IJ)/RR
06465      UZJ1(IJ)=UZJ1(IJ)/RR
06470      RR=SQRT(UXJ2(IJ)**2+UYJ2(IJ)**2+UZJ2(IJ)**2)
06475      UXJ2(IJ)=UXJ2(IJ)/RR
06480      UYJ2(IJ)=UYJ2(IJ)/RR
06485      UZJ2(IJ)=UZJ2(IJ)/RR
06490 95 CONTINUE
06495      RETURN
06500 980 WRITE(6,981)
06505 981 FORMAT(/, ' **** ERROR IN WIRE INPUT')
06510      STOP
06515 990 WRITE(6,991)
06520 991 FORMAT(/, ' **** ERROR IN JUNCTION INPUT')
06525      STOP
06530      END
07190C -----
07200      SUBROUTINE VWIRE(NWJV,VW)
07210C
07220C READ WIRE AND JUNCTION VOLTAGES.
07230C
07240      DIMENSION IW(10)
07250      COMPLEX EW(10)
07260      COMPLEX VW(1)

```

```

C7270      COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
C7280      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RACD(10)
C7290      K=LW+NJ
C7300      DO 25 I=1,K
C7310 25    VW(I)=C.C
C7320      WRITE(6,1)
C7330 1     FORMAT(1H1)
C7340      READ(5,2) NWJV
C7350 2     FORMAT(I3)
C7360      IF(NWJV.EQ.0) RETURN
C7370      READ(5,3) (IW(I),I=1,NWJV)
C7380 3     FORMAT(10I8)
C7390      READ(5,4) (EW(I),I=1,NWJV)
C7395 4     FORMAT(10F8.4)
C7400      WRITE(6,5)
C7410 5     FORMAT(' WIRE VOLTAGES',//,' WIRE INDEX',13X,'VOLTAGE')
C7420      DO 50 I=1,NWJV
C7430      WRITE(6,6) IW(I),EW(I)
C7440 6     FORMAT(1X,I6,9X,2F10.4)
C7450      VW(IW(I))=EW(I)
C7460 50    CONTINUE
C7630      RETURN
C7640      END
-----
C7650C     SUBROUTINE SLOTIN
C7660
C7670C     READ SLOT ANTENNA INPUTS.
C7680C
C7690C     COMPLEX EO
C7700      COMMON /PLOT1/ NPLOT,XPLOT(200),YPLOT(200),ZPLOT(200),ISYM(200)
C7710      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
C7720      COMMON /SLOT1/ NSA,IS(20),ZO(20),Z1(20)
C7730      COMMON /SLOT2/ EO(20),TEXC(20),ZEXC(20)
C7740 16    READ(5,16) NSA
C7750      FORMAT(I3)
C7760 3     WRITE(6,3) NSA
C7770      FORMAT(/,26HNUMBER OF SLOT ANTENNAS =,I3)
C7780      IF(NSA.EQ.0) RETURN
C7790      READ(5,54) (IS(K),K=1,NSA)
C7800 54    FORMAT(10I8)
C7800C     NOTE -L <= ZO(K) < Z1(K) <= L
C7810      READ(5,53) (ZO(K),K=1,NSA)
C7820 53    FORMAT(10F8.4)
C7830      READ(5,53) (Z1(K),K=1,NSA)
C7840      READ(5,53) (EO(K),K=1,NSA)
C7850C     T-EXCITATION.
C7860      READ(5,53) (TEXC(K),K=1,NSA)
C7870C     Z-EXCITATION.
C7880      READ(5,53) (ZEXC(K),K=1,NSA)
C7890      WRITE(6,4) ((K,IS(K),ZO(K),Z1(K),EO(K),TEXC(K),ZEXC(K)),K=1,NSA)
C7900 4     FORMAT(/,50H ANTENNA NO. IS ZO Z1 EO,
C7910      1 11X,14HTEXC ZEXC,/, (I9,I8,6F10.4))

```

```

07911C UPDATE PLOT ARRAY.
07912 DO 150 I=1,NSA
07913 NPLOT=NPLOT+1
07914 IND=IS(I)
07915 XPLOT(NPLOT)=XB(2*IND+1)
07916 YPLOT(NPLOT)=YB(2*IND+1)
07917 ZPLOT(NPLOT)=ZO(I)
07918 ISYM(NPLOT)='S'
07919 NPLOT=NPLOT+1
07920 XPLOT(NPLOT)=XB(2*IND+1)
07921 YPLOT(NPLOT)=YB(2*IND+1)
07922 ZPLOT(NPLOT)=ZI(I)
07923 ISYM(NPLOT)='S'
07924 150 CONTINUE
07925 RETURN
07930 END
07940C -----
07950 SUBROUTINE ORDER(MORD)
07960C DETERMINE THE ORDER IN WHICH THE Y MATRIX WAS GENERATED, AND
07970C CHECK FOR DIMENSIONAL CONSISTENCY BETWEEN THE Y MATRIX DATA FILE,
07980C AND THE INPUT DATA FILE.
07990 DIMENSION MORD(2)
08000 COMMON /BOT1/ NMODE,NPT,NBAND
08010 COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
08020 COMMON /CAP1/ NC,XC,YC,ZC(2)
08025 COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
08030 COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
08032 1 XW1(100),YW1(100),ZW1(100)
08034 COMMON /WIRE3/ NW,INDW(6),RADW(5)
08040 COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
08042 COMMON /EDG1/ NE,ZE(2),ZBE(10)
08050 READ(1) NMODE1,NP1,NC1,NPR1,NE1,NW1,NPW1,NJ1
08052 IF(NMODE.NE.NMODE1) GO TO 500
08054 IF(NP.NE.NP1) GO TO 500
08056 IF(NC.NE.NC1) GO TO 500
08058 IF(NPR.NE.NPR1) GO TO 500
08060 IF(NE.NE.NE1) GO TO 500
08062 IF(NW.NE.NW1) GO TO 500
08064 IF(NPW.NE.NPW1) GO TO 500
08066 IF(NJ.NE.NJ1) GO TO 500
08120C MORD SPECIFIES THE ORDER IN WHICH CAPS AND/OR WIRES HAVE BEEN ADDED
08130C TO THE BOT. INDEX 1 REFERS TO CAPS AND INDEX 2 REFERS TO WIRES.
08140C MORD(I)=C IF I HAS NOT BEEN ADDED
08150C MORD(I)=N IF I HAS BEEN ADDED TO THE SYSTEM MATRIX, AND
08160C IS LOCATED AT PSEUDO MODE NUMBER N.
08170 READ(1) MORD
08220 WRITE(6,4) MORD
08230 4 FORMAT(' THE Y MATRIX CONTAINS THE FOLLOWING ADDITIONS ( 0 IF ',
08240 1 ' NOT INCLUDED, OR CORRESPONDING MODE NUMBER IF PRESENT) : ',//
08250 2 ' 1X, 'CAPS',I4,', 10X, 'WIRES',I3)
08260 RETURN
08270 500 WRITE(6,1)

```

```

08280 1  FORMAT(//,' *** ERROR ***  INPUT PARAMETERS DO NOT CHECK WITH',
08290 1  ' THE PARAMETERS IN THE Y MATRIX FILE',//,
08300 2  ' 6X, ' NMODE NP NC NPR NE NW NPW NJ')
08310 2  WRITE(6,2) NMODE,NP,NC,NPR,NE,NW,NPW,NJ
08320 2  FORMAT(' INPUT ',8I6)
08330 2  WRITE(6,3) NMODE1,NP1,NC1,NPR1,NE1,NW1,NPW1,NJ1
08340 3  FORMAT(' Y FILE ',8I6)
08350 3  STOP
08360 3  END
08370C -----
08380 3  SUBROUTINE VBOT(M,VB)
08390C
08400C COMPUTE BCT VOLTAGE FOR MODE M.
08410C
08420 3  COMPLEX U,A3,A4,VB(1)
08430 3  COMPLEX EO
08440 3  COMMON /SLOT1/ NSA,IS(20),ZO(20),Z1(20)
08450 3  COMMON /SLOT2/ EO(20),TEXC(20),ZEXC(20)
08460 3  COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
08470 3  COMMON /BOT3/ DH(82),SV(82),CV(82)
08480 3  DATA U,PI /(C.,1.),3.14159265/
08490 3  LS=NP-3
08500 3  NM=LS/2
08510 3  DO 5C I=1,LS
08520 3  VB(I)=0.0
08530 3  DO 7C K=1,NSA
08540 3  I=IS(K)
08550 3  I1=I+NM
08560 3  J=2*(I-1)
08570 3  DH4=DH(J+1)+DH(J+2)+DH(J+3)+DH(J+4)
08580 3  A3=Z1(K)-ZO(K)
08590 3  IF(M.EQ.0) GO TO 66
08600 3  A3=-U*BL/M/PI*(CEXP(U*M*PI*Z1(K)/BL)-CEXP(U*M*PI*ZO(K)/BL))
08610 3  A4=A3-(-1)**M*(Z1(K)-ZO(K))
08620 3  VB(I)=VB(I)+EO(K)*DH4/2.*A3*TEXC(K)
08630 3  VB(I1)=VB(I1)+EO(K)*DH4/2.*A4*ZEXC(K)
08640 3  CONTINUE
08650 3  RETURN
08660 3  END
08670C *****
08680 3  SUBROUTINE RBT(RBT,RBP,NT,TH,PHI)
08690C *****
08700C COMPUTE BOT TRANSFER MATRICES FOR THETA AND PHI POLARIZATION.
08710 3  COMPLEX RBT(1),RBP(1),A6,U
08720 3  COMMON /WAVE/ BK
08730 3  COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
08740 3  COMMON /BOT3/ DH(82),SV(82),CV(82)
08750 3  COMMON /BOT5/ T(160),TP(160),TZ(160)
08760 3  DIMENSION TH(1),PHI(1)
08770 3  DATA PI,U /3.14159265,(0.,1.)/
08780 3  DATA DTOR /0.017453292/
08790 3  LS=NP-3

```

```

08800      NM=LS/2
08810      DO 156 K=1,NT
08820      K1=(K-1)*LS
08830      CS=COS(TH(K)*DTOR)
08840      SN=SIN(TH(K)*DTOR)
08850      CSPHI=COS(PHI(K)*DTOR)
08860      SNPHI=SIN(PHI(K)*DTOR)
08870      DO 300 J=1,NM
08880      J1=J+K1
08890      J2=J1+NM
08900      RBT(J1)=0.
08910      RBT(J2)=0.
08920      RBP(J1)=0.
08930      RBP(J2)=0.
08940C  CALCULATION OF TRANSFER MATRICES (EQS. 38-41).
08950C  NOTE, ONLY THE MODE INDEPENDENT PORTION IS COMPUTED HERE.
08960      DO 301 I=1,4
08970      I1=2*(J-1)+I
08980      I4=4*(J-1)+I
08990      A6=CEXP(U*BK*SN*(XB1(I1)*CSPHI+YB1(I1)*SNPHI))
09000C  THETA POLARIZED (EQS. 38-39).
09010      RBT(J1)=RBT(J1)+DH(I1)*T(I4)*(CV(I1)*CSPHI+SV(I1)*SNPHI)*A6
09020      RBT(J2)=RBT(J2)+DH(I1)*TZ(I4)*A6
09030C  PHI POLARIZED (EQS. 40-41).
09040      RBP(J1)=RBP(J1)+DH(I1)*T(I4)*(SV(I1)*CSPHI-CV(I1)*SNPHI)*A6
09050      301 CONTINUE
09060      RBT(J1)=CS*RBT(J1)
09070      RBT(J2)=-SN*RBT(J2)
09080      300 CONTINUE
09090      156 CONTINUE
09100      RETURN
09110      END
09111C -----
09112      SUBROUTINE RCAP(RCT,RCP,NT,TH,PHI)
09113C
09114C  COMPUTE CAP TRANSFER MATRICES FOR THETA AND PHI POLARIZATION.
09115C
09116      COMPLEX RCT(1),RCP(1),A6,U
09117      COMMON /WAVE/ BK
09118      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
09119      COMMON /BOT3/ DH(82),SV(82),CV(82)
09120      COMMON /BOT5/ T(160),TP(160),TZ(160)
09121      COMMON /CAP1/ NC,XC,YC,ZC(2)
09122      COMMON /CAP2/ NPR,RHQC(21),RHQC1(20),DRHQC(20)
09123      COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
09124      COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
09125      COMMON /EDG1/ NE,ZE(2),ZBE(10)
09126      COMMON /EDG2/ TCE(10),TPCE(10),TBE(10),TPBE(10)
09127      DIMENSION TH(1),PHI(1)
09128      DATA PI,U /3.14159265,(0.,1.)/
09129      DATA DTOR /0.017453292/
09130      NM=(NP-3)/2

```

```

09131      LR=(NPR-3)/2
09132      LC=NC*NM*LR
09133      LE=NE*NM
09134      DO 156 K=1,NT
09135      K1=(K-1)*I2+LC+LE
09136      CS=COS(TH(K)*DTOR)
09137      SN=SIN(TH(K)*DTOR)
09138      CSPHI=COS(PHI(K)*DTOR)
09139      SNPHI=SIN(PHI(K)*DTOR)
09140      J1=K1
09141      DO 300 JC=1,NC
09142      DO 300 JP=1,NM
09143      DO 300 JR=1,LR
09144      J1=J1+1
09145      J2=J1+LC
09146      RCT(J1)=0.
09147      RCT(J2)=0.
09148      RCP(J1)=0.
09149      RCP(J2)=0.
09150C  CALCULATION OF TRANSFER MATRICES.
09151      DO 301 IIP=1,4
09152      IP1=2*(JP-1)+IIP
09153      IP4=4*(JP-1)+IIP
09154      DO 301 IIR=1,4
09155      IR1=2*(JR-1)+IIR
09156      IR4=4*(JR-1)+IIR
09157      XX=XC+RHOC1(IR1)*RC1(IP1)*CPC(IP1)
09158      YY=YC+RHOC1(IP1)*RC1(IP1)*SPC(IP1)
09159      ZZ=ZC(JC)
09160      A6=CEXP(U*BK*(SN*(XX*CSPHI+YY*SNPHI)+ZZ*CS))
09161      AI=AC(IP1)*ABS(RHOC(IR1+1)**2-RHOC(IR1)**2)
09162C  THETA POLARIZED.
09163      RCT(J1)=RCT(J1)+AI*(CV(IP1)*CSPHI+SV(IP1)*SNPHI)*
09164      1 T(IP4)*TCT(IR4)*A6
09165      RCT(J2)=RCT(J2)+AI*(CPC(IP1)*CSPHI+SPC(IP1)*SNPHI)*
09166      1 TZ(IP4)*TCR(IR4)*A6
09167C  PHI POLARIZED.
09168      RCP(J1)=RCP(J1)+AI*(SV(IP1)*CSPHI-CV(IP1)*SNPHI)*
09169      1 T(IP4)*TCT(IR4)*A6
09170      RCP(J2)=RCP(J2)+AI*(SPC(IP1)*CSPHI-CPC(IP1)*SNPHI)*
09171      1 TZ(IP4)*TCR(IR4)*A6
09172 301 CONTINUE
09173      RCT(J1)=CS*RCT(J1)
09174      RCT(J2)=CS*RCT(J2)
09175 300 CONTINUE
09176      IF(NE.EQ.0) GO TO 156
09177C  EDGE TRANSFER MATRICES FOLLOW.
09178      J2=2*LC+K1
09179      DO 400 JC=1,NC
09180      DO 400 JP=1,NM
09181      J2=J2+1
09182      RCT(J2)=C.

```

```

09183 RCP(J2)=0.
09184 DO 401 IIP=1,4
09185 IP1=2*(JP-1)+IIP
09186 IP4=4*(JP-1)+IIP
09187 DO 401 IIRZ=1,2
09188 IR1=2*LR+IIRZ
09189 IR4=2*(JC-1)+IIRZ
09190 IZ1=2*(JC-1)+IIRZ
09191 IZ4=2*(JC-1)+IIRZ
09192 XX=XC+RHOC1(IR1)*RC1(IP1)*CPC(IP1)
09193 YY=YC+RHOC1(IR1)*RC1(IP1)*SPC(IP1)
09194 ZZ=ZC(JC)
09195 A6=CEXP(U*BK*(SN*(XX*CSPHI+YY*SNPHI)+ZZ*CS))
09196 AI=AC(IP1)*ABS(RHOC(IR1+1)**2-RHOC(IR1)**2)
09197C EDGE(CAP)
09198C THETA POLARIZED.
09199 RCT(J2)=RCT(J2)+CS*AI*(CPC(IP1)*CSPHI+SPC(IP1)*SNPHI)*
09200 1 TZ(IP4)*TCE(IR4)*A6
09201C PHI POLARIZED.
09202 RCP(J2)=RCP(J2)+AI*(SPC(IP1)*CSPHI-CPC(IP1)*SNPHI)*
09203 1 TZ(IP4)*TCE(IR4)*A6
09204 XX=XB1(IP1)
09205 YY=YB1(IP1)
09206 ZZ=ZB1(IZ1)
09207 A6=CEXP(U*BK*(SN*(XX*CSPHI+YY*SNPHI)+ZZ*CS))
09208 AI=AI(IP1)*ABS(ZE(JC)-ZC(JC))/2.
09209C EDGE(BOT)
09210C THETA POLARIZED.
09211 RCT(J2)=RCT(J2)-SN*AI*TZ(IP4)*TBE(IZ4)*A6
09212 401 CONTINUE
09213 400 CONTINUE
09214 156 CONTINUE
09215 RETURN
09216 END
09217C -----
09218C SUBROUTINE RWIRE(RWT,RWP,NT,TH,PHI)
09219C
09220C COMPUTE WIRE-JUNCTION TRANSFER MATRICES.
09221C
09222C COMPLEX RWT(1),RWP(1),A6,U
09223C COMMON /WAVE/ WK
09224C COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
09225C 1 XW1(100),YW1(100),ZW1(100)
09226C COMMON /WIRE2/ DHW(100),DXW(100),DYW(100),DZW(100)
09227C COMMON /WIRE3/ NW,INDW(6),RADW(5)
09228C COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
09229C COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
09230C COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
09231C COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
09232C COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
09233C COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
09234C DIMENSION TH(1),PHI(1)

```

```

10350 DATA PI,U /3.14159265,(0.,1.)/
10360 DATA DTOR /0.017453292/
10370 LWJ=LW+NJ
10380 DO 156 K=1,NT
10390 K1=(K-1)*LWJ
10400 CS=COS(TH(K)*DTOR)
10410 SN=SIN(TH(K)*DTOR)
10420 CSPHI=COS(PHI(K)*DTOR)
10430 SNPHI=SIN(PHI(K)*DTOR)
10440 DO 300 J=1,LW
10450 J1=J+K1
10460 RWT(J1)=0.
10470 RWP(J1)=0.
10480 DO 301 I=1,4
10490 I1=INDTM(J)+I-1
10500 I4=4*(J-1)+I
10510 A6=CEXP(U*BK*(SN*(XW1(I1)*CSPHI+YW1(I1)*SNPHI)+ZW1(I1)*CS))
10520 RWT(J1)=RWT(J1)+(CS*(DXW(I1)*CSPHI+DYW(I1)*SNPHI)-
10530 1 DZW(I1)*SN)*TW(I4)*A6
10540 RWP(J1)=RWP(J1)+(-DXW(I1)*SNPHI+DYW(I1)*CSPHI)*TW(I4)*A6
10550 301 CONTINUE
10560 300 CONTINUE
10570 IF(NJ.EQ.0) GO TO 156
10580 J1=LW+K1
10590 DO 400 J=1,NJ
10600 J1=J1+1
10610 RWT(J1)=0.
10620 RWP(J1)=0.
10630 DO 401 I=1,2
10640 I1=INDTJ(J)+I-1
10650 I2=2*(J-1)+I
10660 A6=CEXP(U*BK*(SN*(XW1(I1)*CSPHI+YW1(I1)*SNPHI)+ZW1(I1)*CS))
10670 RWT(J1)=RWT(J1)+(CS*(DXW(I1)*CSPHI+DYW(I1)*SNPHI)-
10680 1 DZW(I1)*SN)*TJ(I2)*A6
10690 RWP(J1)=RWP(J1)+(-DXW(I1)*SNPHI+DYW(I1)*CSPHI)*TJ(I2)*A6
10700 401 CONTINUE
10710 DO 402 JJ=1,4
10720 ALPHA=PI*((JJ-1)/2.+0.25)
10730 SA=SIN(ALPHA)
10740 CA=COS(ALPHA)
10750 RA=(RADJ(J)+RADD(J))/2.
10760 XA=XJ(J)+RA*(CA*UXJ1(J)+SA*UXJ2(J))
10770 YA=YJ(J)+RA*(CA*UYJ1(J)+SA*UYJ2(J))
10780 ZA=ZJ(J)+RA*(CA*UZJ1(J)+SA*UZJ2(J))
10790 DT=CA*(UXJ1(J)*CS+CSPHI+UYJ1(J)*CS+SNPHI-UZJ1(J)*SN)+
10800 1 SA*(UXJ2(J)*CS+CSPHI+UYJ2(J)*CS+SNPHI-UZJ2(J)*SN)
10810 DP=CA*(-UXJ1(J)*SNPHI+UYJ1(J)*CSPHI)+
10820 1 SA*(-UXJ2(J)*SNPHI+UYJ2(J)*CSPHI)
10830 RR=(RADJ(J)-RADD(J))/2.
10840 A6=CEXP(U*BK*(SN*(XA*CSPHI+YA*SNPHI)+ZA*CS))
10850 RWT(J1)=RWT(J1)+RR/4.*DT*A6
10860 RWP(J1)=RWP(J1)+RR/4.*DP*A6

```

```

10870 402 CONTINUE
10880 400 CONTINUE
10890 136 CONTINUE
10900 RETURN
10910 END
10920C *****
10930 SUBROUTINE LPCUR(NMODE,NM,CB,LW,NJ,CW,LC,LE,CC)
10940C *****
10950C LIST AND PLOT BOT, WIRE, AND CAP CURRENTS.
10960 COMPLEX U,A3,A4,CB(1),CW(1),CC(1)
10970 DIMENSION TMAG(41),TPHS(41),ZMAG(41),ZPHS(41)
10980 DATA PI,U /3.14159265,(0.,1.)/
10990 WRITE(6,3)
11000 3 FORMAT(16H1 BOT CURRENTS)
11010 NM=2*NMODE-1
11020 LS=NM+2
11030 DO 100 NM=1,KM
11040 M=NM-NMODE
11050 I1=(NM-1)*LS+1
11060 I2=I1+NM-1
11070 WRITE(6,1) M,(CB(I),I=I1,I2)
11080 1 FORMAT(/,29H T-DIRECTED CURRENTS FOR MODE,I3,/,4(2E12.4,5X))
11090 I1=I2+1
11100 I2=I1+NM-1
11110 WRITE(6,2) M,(CB(I),I=I1,I2)
11120 2 FORMAT(/,29H Z-DIRECTED CURRENTS FOR MODE,I3,/,4(2E12.4,5X))
11130 100 CONTINUE
11140 DO 320 I=1,NM
11150 DO 310 K=1,41
11160 A3=0.0
11170 A4=0.0
11180 ZE=-1.0+(K-1)*0.05
11190 M=-NMODE
11200 I1=I-LS
11210 DO 305 NM=1,KM
11220 M=M+1
11230 I1=I1+LS
11240 I2=I1+NM
11250C SUM THE T AND Z DIRECTED CURRENTS ON TRIANGLE FUNCTION I AT Z=ZE,
11260C (EQ. 8).
11270 A3=A3+CB(I1)*CEXP(U*M*PI*ZE)
11280 A4=A4+CB(I2)*ICEXP(U*M*PI*ZE)-((-1)**M)
11290 305 CONTINUE
11300 TMAG(K)=CABS(A3)/2.
11310 TPHS(K)=ATAN2(AIMAG(A3),REAL(A3))*180./PI
11320 ZMAG(K)=CABS(A4)/2.
11330 ZPHS(K)=ATAN2(AIMAG(A4),REAL(A4))*180./PI
11340 310 CONTINUE
11350 WRITE(6,301)
11360 301 FORMAT(1H1)
11370 WRITE(6,302) I
11380 302 FORMAT(41H T-DIRECTED CURRENTS ON TRIANGLE FUNCTION,I3,/)

```

```

11390C PLOT MAGNITUDE AND PHASE.
11400 CALL PLOTG(TMAG,TPHS)
11410 WRITE(6,304) I
11420 304 FORMAT(///,41H Z-DIRECTED CURRENTS ON TRIANGLE FUNCTION,I3,/)
11430C PLOT MAGNITUDE AND PHASE.
11440 CALL PLOTG(ZMAG,ZPHS)
11450 320 CONTINUE
11460 IF(LW.EQ.0) GO TO 500
11470 WRITE(6,4)
11480 4 FORMAT(//,16H WIRE CURRENTS,/)
11490 WRITE(6,5) (CW(I),I=1,LW)
11500 5 FORMAT(4(2E12.4,5X))
11510 500 CONTINUE
11520 IF(NJ.EQ.0) GO TO 600
11530 I1=LW+1
11540 I2=LW+NJ
11550 WRITE(6,6)
11560 6 FORMAT(//,20H JUNCTION CURRENTS,/)
11570 WRITE(6,5) (CW(I),I=I1,I2)
11580 600 CONTINUE
11590 IF(LC.EQ.0) RETURN
11600 WRITE(6,7)
11610 7 FORMAT(//,15H CAP CURRENTS,/)
11620 WRITE(6,8)
11630 8 FORMAT(//,11H T-DIRECTED)
11640 WRITE(6,5) (CC(I),I=1,LC)
11650 WRITE(6,9)
11660 9 FORMAT(//,11H R-DIRECTED)
11670 I1=LC+1
11680 I2=2*LC
11690 WRITE(6,5) (CC(I),I=I1,I2)
11700 IF(LE.EQ.0) RETURN
11710 10 FORMAT(//,14H EDGE CURRENTS)
11720 I1=2*LC+1
11730 I2=2*LC+LE
11740 WRITE(6,5) (CC(I),I=I1,I2)
11750 RETURN
11760 END
-----
11770C SUBROUTINE PLOTG(Y1,Y2)
11780 *****
11790C
11800C WRITTEN 2/14/74 BY J. M. PUTNAM DEPT 220 X23877
11810C
11820C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
11830C N IS THE NUMBER OF POINTS TO BE PLOTTED.
11840C NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
11850C NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
11860C NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
11870C
11880C REAL Y1(41),Y2(41),HEAD(10)

```

```

11900      INTEGER LINE(101),BLANK,STAR,PLUS
11910      DATA BLANK,STAR,PLUS /1H,1H*,1H+/
11920      N=41
11930      NR=21
11940      NC=41
11950      N10=(NC-1)/10
11960      WRITE(6,500)
11970 500    FORMAT(12H * MAGNITUDE,45X,7H+ PHASE)
11980      WRITE(6,504)
11990      XMIN=-1.0
12000      XMAX=1.0
12010      YMIN1=Y1(1)
12020      YMAX1=Y1(1)
12030      YMIN2=-180.0
12040      YMAX2=180.0
12050      DO 6 I=1,N
12060      IF(Y1(I).LT.YMIN1) YMIN1=Y1(I)
12070      IF(Y1(I).GT.YMAX1) YMAX1=Y1(I)
12080 6      CONTINUE
12090      DO 5 I=1,N10
12100      Z=I
12110 5      HEAD(I)=(XMAX-XMIN)*Z/N10+XMIN
12120      DY1=(YMAX1-YMIN1)/(NR-1)
12130      DY2=18.0
12140      Z1=YMAX1+DY1
12150      Z2=YMAX2+DY2
12160      YL1=Z1-DY1/2.
12170      YL2=Z2-DY2/2.
12180      DO 7 J=1,NR
12190      DO 8 K=1,NC
12200 8      LINE(K)=BLANK
12210      Z1=Z1-DY1
12220      Z2=Z2-DY2
12230      YU1=YL1
12240      YU2=YL2
12250      YL1=Z1-DY1/2.
12260      YL2=Z2-DY2/2.
12270      DO 9 I=1,N
12280      X=XMIN+(XMAX-XMIN)*(I-1)/FLOAT(N-1)
12290      IF(Y2(I).GE.YU2) GO TO 10
12300      IF(Y2(I).LT.YL2) GO TO 10
12310      K=(X-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
12320      IF(K.GT.NC) K=NC
12330      LINE(K)=PLUS
12340 10      IF(Y1(I).GE.YU1) GO TO 9
12350      IF(Y1(I).LT.YL1) GO TO 9
12360      K=(X-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
12370      IF(K.GT.NC) K=NC
12380      LINE(K)=STAR
12390 9      CONTINUE
12400      WRITE(6,508) Z1,(LINE(K),K=1,NC),Z2
12410 7      CONTINUE

```

```

12420      WRITE(6,504)
12430      WRITE(6,3002)
12440      WRITE(6,507) XMIN,(HEAD(I),I=1,N10)
12450      RETURN
12460      504 FORMAT (1X,14(1H-),1H,8(5H----.),1H-)
12470      507 FORMAT(3X,3HZ/L,5X,11(G10.2))
12480      508 FORMAT(1X,6I2,3(1X,1H),4I1,1HI,F7.1)
12490      3002 FORMAT(15X,1H,476X,1H)
12500      END
-----
12510C SUBROUTINE NEAR8(XTEST,YTEST,ZTEST,ZM)
12520C CALCULATE THE M-TH MODAL CURRENT COEFFICIENTS FOR THE NEAR FIELD
12530C ANALYSIS (EQS 67 & 71).
12540C COMPLEX U,A3,ZM(1)
12550      COMPLEX GT(82),GZ(82),G1T(82),G1Z(82),H1T(82)
12560      COMMON /WAVE/ BK
12570      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
12580      COMMON /BOT3/ DH(82),SV(82),CV(82)
12590      COMMON /BOT5/ T(160),TP(160),TZ(160)
12600      COMMON /INT/ M,RHO2,ZP
12610      EXTERNAL FUNCT,FUNCZ,FUNCT,FUNC1Z,FUNC2T
12620      DATA PI,U /3.14159265,(0.,1.)/
12630      BKL=BK*BL
12640      KG=NP-1
12650      NM=(NP-3)/2
12660      ETA=376.707
12670      I1=0
12680C COMPUTATION OF GREEN FUNCTION KERNELS G, H0, AND H1.
12690      DO 16 J=1,KG
12700      I1=I1+1
12710      YY=YB1(J)-YTEST
12720      XX=XB1(J)-XTEST
12730      RHO2=XX*XX+YY*YY
12740      ZP=ZTEST
12750      CALL CSIMP(FUNCT,-BL,BL,0.05,10,A3,GT(I1),K,IER)
12760      GT(I1)=GT(I1)/4./PI
12770      CALL CSIMP(FUNCZ,-BL,BL,0.05,10,A3,GZ(I1),K,IER)
12780      GZ(I1)=GZ(I1)/4./PI
12790      CALL CSIMP(FUNCT,-BL,BL,0.05,10,A3,G1T(I1),K,IER)
12800      G1T(I1)=G1T(I1)/4./PI
12810      CALL CSIMP(FUNC1Z,-BL,BL,0.05,10,A3,G1Z(I1),K,IER)
12820      G1Z(I1)=G1Z(I1)/4./PI
12830      CALL CSIMP(FUNC2T,-BL,BL,0.05,10,A3,H1T(I1),K,IER)
12840      H1T(I1)=H1T(I1)/4./PI
12850      16 CONTINUE
12860C COMPUTATION OF M-TH MODAL CURRENT COEFFICIENTS (EQS. 67 & 71).
12870      DO 30 J=1,NM
12880      JL=(J-1)*6
12890      J3=(J-1)*4
12900      J1=2*(J-1)
12910C ELECTRIC FIELD.
12920      L1=JL+1
12930

```

```

12940      L2=L1+NM*6
12950      L3=L1+1
12960      L4=L2+1
12970      L5=L1+2
12980      L6=L2+2
12990      ZM(L1)=0.
13000      ZM(L2)=0.
13010      ZM(L3)=0.
13020      ZM(L4)=0.
13030      ZM(L5)=0.
13040      ZM(L6)=0.
13050C    MAGNETIC FIELD.
13060      K1=L1+3
13070      K2=L2+3
13080      K3=L1+4
13090      K4=L2+4
13100      K5=L1+5
13110      K6=L2+5
13120      ZM(K1)=0.
13130      ZM(K2)=0.
13140      ZM(K3)=0.
13150      ZM(K4)=0.
13160      ZM(K5)=0.
13170      ZM(K6)=0.
13180      DO 70 JJ=1,4
13190      J2=J1+JJ
13200      J7=J3+JJ
13210      J4=J2
13220      XX=XB1(J2)-XTEST
13230      YY=YB1(J2)-YTEST
13240C    ELECTRIC FIELD.
13250C    UX COMPONENT.
13260      ZM(L1)=ZM(L1)+DH(J2)*(CV(J2)*T(J7)*GT(J4)+
13270      1 XX*TP(J7)*G1T(J4)/BK/BK)
13280      ZM(L2)=ZM(L2)+DH(J2)*XX*TZ(J7)*G1T(J4)
13290C    UY COMPONENT.
13300      ZM(L3)=ZM(L3)+DH(J2)*(SV(J2)*T(J7)*GT(J4)+
13310      1 YY*TP(J7)*G1T(J4)/BK/BK)
13320      ZM(L4)=ZM(L4)+DH(J2)*YY*TZ(J7)*G1T(J4)
13330C    UZ COMPONENT.
13340      ZM(L5)=ZM(L5)+DH(J2)*TP(J7)*H1T(J4)
13350      ZM(L6)=ZM(L6)+DH(J2)*TZ(J7)*(GZ(J4)+
13360      1 U*M*PI*H1T(J4)/BK/BK)
13370C    MAGNETIC FIELD.
13380C    UX COMPONENT.
13390      ZM(K1)=ZM(K1)-DH(J2)*SV(J2)*T(J7)*H1T(J4)
13400      ZM(K2)=ZM(K2)+DH(J2)*YY*TZ(J7)*G1Z(J4)
13410C    UY COMPONENT.
13420      ZM(K3)=ZM(K3)+DH(J2)*CV(J2)*T(J7)*H1T(J4)
13430      ZM(K4)=ZM(K4)-DH(J2)*XX*TZ(J7)*G1Z(J4)
13440C    UZ COMPONENT.
13450      ZM(K5)=ZM(K5)-DH(J2)*T(J7)*(YY*CV(J2)-

```

```

13460      1 XX*SV(J2))*GIT(J4)
13470      70 CONTINUE
13480C    MULTIPLY BY CONSTANTS.
13490      ZM(L1)=U*BK*ETA*ZN(L1)
13500      ZM(L2)=-M*PI*ETA/BKL*ZN(L2)
13510      ZM(L3)=U*BK*ETA*ZN(L3)
13520      ZM(L4)=-M*PI*ETA/BKL*ZN(L4)
13530      ZM(L5)=U*ETA/BK*ZN(L5)
13540      ZM(L6)=U*BK*ETA*ZN(L6)
13550      30 CONTINUE
13560      RETURN
13570      END
-----
13580C    SUBROUTINE NEARC(XTEST,YTEST,ZTEST,ZM)
13590C    CALCULATE THE CAP CURRENT COEFFICIENTS FOR THE NEAR FIELD ANALYSIS.
13600      COMPLEX U,A3,A4,ZN(1)
13610      COMMON /WAVE/ BK
13620      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
13630      COMMON /BOT3/ DH(82),SV(82),CV(82)
13640      COMMON /BOT5/ T(160),TP(160),TZ(160)
13650      COMMON /CAP1/ NC,XC,YC,ZC(2)
13660      COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
13670      COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
13680      COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
13690      DATA PI,U /3.14159265,(0.,1.)/
13700      NM=(NP-3)/2
13710      LR=(NPR-3)/2
13720      LC=NC+NM+LR
13730      BK2=BK*BK
13740      ETA=376.707
13750      J=0
13760C    NOTE, EDGE COEFFICIENTS ARE NOT INCLUDED.
13770      DO 30 JC=1,NC
13780      DO 30 JP=1,NM
13790      DO 30 JR=1,LR
13800      J=J+1
13810      JL=(J-1)*6
13820C    ELECTRIC FIELD.
13830      L1=JL+1
13840      L2=L1+LC*6
13850      L3=L1+1
13860      L4=L2+1
13870      L5=L1+2
13880      L6=L2+2
13890      ZM(L1)=0.
13900      ZM(L2)=0.
13910      ZM(L3)=0.
13920      ZM(L4)=0.
13930      ZM(L5)=0.
13940      ZM(L6)=0.
13950C    MAGNETIC FIELD.
13960      K1=L1+3
13970

```

```

3980 K2=L2+3
3990 K3=L1+4
4000 K4=L2+4
4010 K5=L1+3
4020 K6=L2+5
4030 ZH(K1)=0.
4040 ZH(K2)=0.
4050 ZH(K3)=0.
4060 ZH(K4)=0.
4070 ZH(K5)=0.
4080 ZH(K6)=0.
4090 JP2=2*(JP-1)
4100 JP7=4*(JP-1)
4110 DO 70 JJP=1,4
4120 JP2=JP2+1
4130 JP7=JP7+1
4140 JR2=2*(JR-1)
4150 JR7=4*(JR-1)
4160 DO 70 JJR=1,4
4170 JR2=JR2+1
4180 JR7=JR7+1
4190 AJ=AC(JP2)*ABS(RHOC(JR2+1)**2-RHOC(JR2)**2)
4200 RJ=RHOC1(JR2)*RC1(JP2)
4210 XX=XC+RJ*CPC(JP2)-XTEST
4220 YY=YC+RJ*SPC(JP2)-YTEST
4230 ZZ=ZC(JC)-ZTEST
4240 RR=SQRT(XX*XX+YY*YY+ZZ*ZZ)
4250 A3=CEXP(-U*BK*RR)/4./PI/RR
4260 A4=(1./RR+U*BK)/RR*A3
4270C ELECTRIC FIELD.
4280C UX COMPONENT.
4290 ZH(L1)=ZH(L1)+AJ*TCT(JR7)*(CV(JP2)*T(JP7)+A3+
4300 1 XX/BK2*TP(JP7)/RHOC1(JR2)+A4)
4310 ZH(L2)=ZH(L2)+AJ*TZ(JP7)*(CPC(JP2)*TCR(JR7)+A3+
4320 1 XX/BK2*(TPCR(JR7)/RC1(JP2)+TCR(JR7)/RJ)+A4)
4330C UY COMPONENT.
4340 ZH(L3)=ZH(L3)+AJ*TCT(JR7)*(SV(JP2)*T(JP7)+A3+
4350 1 YY/BK2*TP(JP7)/RHOC1(JR2)+A4)
4360 ZH(L4)=ZH(L4)+AJ*TZ(JP7)*(SPC(JP2)*TCR(JR7)+A3+
4370 1 YY/BK2*(TPCR(JR7)/RC1(JP2)+TCR(JR7)/RJ)+A4)
4380C UZ COMPONENT.
4390 ZH(L5)=ZH(L5)+AJ*ZZ*TP(JP7)/RHOC1(JR2)*TCT(JR7)+A4
4400 ZH(L6)=ZH(L6)+AJ*ZZ*TZ(JP7)+
4410 1 (TPCR(JR7)/RC1(JP2)+TCR(JR7)/RJ)+A4
4420C MAGNETIC FIELD.
4430C UX COMPONENT.
4440 ZH(K1)=ZH(K1)-AJ*ZZ*SV(JP2)*T(JP7)*TCT(JR7)+A4
4450 ZH(K2)=ZH(K2)-AJ*ZZ*SPC(JP2)*TZ(JP7)*TCR(JR7)+A4
4460C UY COMPONENT.
4470 ZH(K3)=ZH(K3)+AJ*ZZ*CV(JP2)*T(JP7)*TCT(JR7)+A4
4480 ZH(K4)=ZH(K4)+AJ*ZZ*CPC(JP2)*TZ(JP7)*TCR(JR7)+A4
4490C UZ COMPONENT.

```

```

14500      ZH(K5)=ZH(K5)-AJ*T(JP7)*TCT(JR7)*
14510      1 (YY*CV(JP2)-XX*SV(JP2))*A4
14520      ZH(K6)=ZH(K6)-AJ*TZ(JP7)*TCR(JR7)*
14530      1 (YY*CPC(JP2)-XX*SPC(JP2))*A4
14540      70 CONTINUE
14550C    MULTIPLY BY CONSTANTS.
14560      ZH(L1)=U*BK*ETA*ZH(L1)
14570      ZH(L2)=U*BK*ETA*ZH(L2)
14580      ZH(L3)=U*BK*ETA*ZH(L3)
14590      ZH(L4)=U*BK*ETA*ZH(L4)
14600      ZH(L5)=U*ETA/BK*ZH(L5)
14610      ZH(L6)=U*ETA/BK*ZH(L6)
14620      30 CONTINUE
14630      RETURN
14640      END
14650C    -----
14660      SUBROUTINE NEARW(XTEST,YTEST,ZTEST,ZH)
14670C    CALCULATE THE WIRE/JUNCTION CURRENT COEFFICIENTS FOR THE NEAR
14680C    FIELD ANALYSIS.
14690      COMPLEX U,A3,A4,ZH(1)
14700      COMMON /WAVE/ BK
14710      COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
14720      1 XW1(100),YW1(100),ZW1(100)
14730      COMMON /WIRE2/ DHW(100),DXW(100),DYW(100),DZW(100)
14740      COMMON /WIRE3/ NW,INDW(6),RADW(5)
14750      COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
14760      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
14770      COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
14780      COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
14790      COMMON /JUNC4/ UXJ(10),UYJ(10),UZJ(10)
14800      COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
14810      COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
14820      DATA PI,U /3.14159265,(0.,1.)/
14830      LWJ=LW+NJ
14840      BK2=BK*BK
14850      ETA=376.707
14855C    NOTE, JUNCTION COEFFICIENTS ARE NOT INCLUDED.
14860      DO 30 J=1,LW
14870      J1=(J-1)*6
14880      J3=(J-1)*4
14890      J1=INDTW(J)-1
14900C    ELECTRIC FIELD.
14910      L1=J1+1
14920      L3=L1+1
14930      L5=L1+2
14940      ZH(L1)=0.
14950      ZH(L3)=0.
14960      ZH(L5)=0.
14970C    MAGNETIC FIELD.
14980      K1=L1+3
14990      K3=L1+4
15000      K5=L1+5

```

```

15010      ZM(K1)=0.
15020      ZM(K3)=C.
15030      ZM(K5)=0.
15040      DO 70 JJ=1,4
15050      J2=J1+JJ
15060      J7=J3+JJ
15070      XX=X(J1,J2)-XTEST
15080      YY=Y(J1,J2)-YTEST
15090      ZZ=Z(J1,J2)-ZTEST
15100      RR=SQRT(XX*XX+YY*YY+ZZ*ZZ)
15110      A3=CEXP(-U*BK*RR)/4./PI/RR
15120      A4=(1./RR+U*BK)/RR*A3
15130      ELECTRIC FIELD.
15140      UX COMPONENT.
15150      ZM(L1)=ZM(L1)+DXW(J2)*TW(J7)*A3+
15160      1 DHW(J2)*XX*TPW(J7)/BK2*A4
15170      UY COMPONENT.
15180      ZM(L3)=ZM(L3)+DYW(J2)*TW(J7)*A3+
15190      1 DHW(J2)*YY*TPW(J7)/BK2*A4
15200      UZ COMPONENT.
15210      ZM(L5)=ZM(L5)+DZW(J2)*TW(J7)*A3+
15220      1 DHW(J2)*ZZ*TPW(J7)/BK2*A4
15230      MAGNETIC FIELD.
15240      UX COMPONENT.
15250      ZM(K1)=ZM(K1)-(DYW(J2)*ZZ-
15260      1 DZW(J2)*YY)*TW(J7)*A4
15270      UY COMPONENT.
15280      ZM(K3)=ZM(K3)-(DXW(J2)*XX-
15290      1 DXW(J2)*ZZ)*TW(J7)*A4
15300      UZ COMPONENT.
15310      ZM(K5)=ZM(K5)-(DXW(J2)*YY-
15320      1 DYW(J2)*XX)*TW(J7)*A4
15330      70 CONTINUE
15340      MULTIPLY BY CONSTANTS.
15350      ZM(L1)=U*BK*ETA*ZM(L1)
15360      ZM(L3)=U*BK*ETA*ZM(L3)
15370      ZM(L5)=U*BK*ETA*ZM(L5)
15380      30 CONTINUE
15390      RETURN
15400      END
15410      *****
15420      SUBROUTINE CSIMP(F,A,B,DEL,IMAX,S11,S,N,IER)
15430      *****
15440      COMPLEX SIMPSON INTEGRATION ROUTINE.
15450      COMPLEX F,S11,S,SUMK
15460      S11=(.0,.0)
15470      S=(.0,.0)
15480      N=0
15490      BA=B-A
15500      IF (BA)20,19,20
15510      19 IER=1
15520
15530

```

```

15540      RETURN
15550 20 IF (DEL) 22, 22, 23
15560 22 IER=2
15570      RETURN
15580 23 IF (IMAX-1) 24, 24, 25
15590 24 IER=3
15600      RETURN
15610 25 X=BA/2, +A
15620      NHALF=1
15630      SUMK=F(X)*BA*2./3.
15640      S=SUMK+(F(A)+F(B))*BA/6.
15650      DO 26 I=2, IMAX
15660          SI1=S
15670          S=(S-SUMK/2.)/2.
15680          NHALF=NHALF*2
15690          ANHLF=NHALF
15700          FRSTX=A+(BA/ANHLF)/2.
15710          SUMK=F(FRSTX)
15720          XK=FRSTX
15730          KLAST=NHALF-1
15740          FINC=BA/ANHLF
15750          DO 26 K=1, KLAST
15760              XK=XK+FINC
15770 26 SUMK=SUMK+F(XK)
15780          SUMK=SUMK*2.*BA/(3.*ANHLF)
15790          S=S+SUMK
15800          IF (CABS(S).EQ.0.0) GO TO 29
15810          IF ((CABS(S-SI1)/CABS(S))-DEL) 29, 28, 28
15820 28 CONTINUE
15830          IER=4
15840          WRITE(6,1) SI1, S
15850 1 FORMAT(/, 29H INTEGRATION DID NOT CONVERGE, //
15860          1 50H THE PREVIOUS AND FINAL VALUES OF THE INTEGRAL ARE,
15870          2 13H RESPECTIVELY, 4E12.3)
15880          GO TO 30
15890 29 IER=0
15900 30 N=2*NHALF
15910      RETURN
15920      END
15930C -----
15940      COMPLEX FUNCTION FUNCT(Z)
15950      COMPLEX U
15960      COMMON /WAVE/ BK
15970      COMMON /BOT2/ NP, BL
15980      COMMON /INT/ M, RHO2, ZP
15990      DATA PI, U /3.14159265, (0., 1.) /
16000      R=SQRT(RHO2*(Z-ZP)**2)
16010      FUNCT=CEXP(U*(M*PI+Z/BL-BK*R))/R
16020      RETURN
16030      END
16040C -----
16050      COMPLEX FUNCTION FUNCZ(Z)

```

```

16060      COMPLEX U
16070      COMMON /WAVE/ BK
16080      COMMON /BOT2/ NP,BL
16090      COMMON /INT/ M,RHO2,ZP
16100      DATA PI,U /3.14159265,(0.,1.)/
16110      R=SQRT(RHO2+(Z-ZP)**2)
16120      FUNCZ=(CEXP(U*M*PI*Z/BL)-(-1)**M)*CEXP(-U*BK*R)/R
16130      RETURN
16140      END
16150C -----
16160      COMPLEX FUNCTION FUNC1(Z)
16170      COMPLEX U
16180      COMMON /WAVE/ BK
16190      COMMON /BOT2/ NP,BL
16200      COMMON /INT/ M,RHO2,ZP
16210      DATA PI,U /3.14159265,(0.,1.)/
16220      R=SQRT(RHO2+(Z-ZP)**2)
16230      FUNC1=CEXP(U*(M*PI*Z/BL-BK*R))*(1./R+U*BK)/R/R
16240      RETURN
16250      END
16260C -----
16270      COMPLEX FUNCTION FUNC1Z(Z)
16280      COMPLEX U
16290      COMMON /WAVE/ BK
16300      COMMON /BOT2/ NP,BL
16310      COMMON /INT/ M,RHO2,ZP
16320      DATA PI,U /3.14159265,(0.,1.)/
16330      R=SQRT(RHO2+(Z-ZP)**2)
16340      FUNC1Z=(CEXP(U*M*PI*Z/BL)-(-1)**M)*CEXP(-U*BK*R)*
16350      1 (1./R+U*BK)/R/R
16360      RETURN
16370      END
16380C -----
16390      COMPLEX FUNCTION FUNC2(Z)
16400      COMPLEX U
16410      COMMON /WAVE/ BK
16420      COMMON /BOT2/ NP,BL
16430      COMMON /INT/ M,RHO2,ZP
16440      DATA PI,U /3.14159265,(0.,1.)/
16450      R=SQRT(RHO2+(Z-ZP)**2)
16460      FUNC2=CEXP(U*(M*PI*Z/BL-BK*R))*(Z-ZP)*(1./R+U*BK)/R/R
16470      RETURN
16480      END
16490C -----
16500      SUBROUTINE PLOT(X,Y,ISYM,N,LX,LY,NC,NR)
16510C -----
16520C
16530C WRITTEN 2/14/74 BY J. M. PUTNAM DEPT 220 X23877
16540C
16550C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
16560C N IS THE NUMBER OF POINTS TO BE PLOTTED.
16570C NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.

```

```

16580C      NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
16590C      NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
16600C
16610      DIMENSION X(1),Y(1),ISYM(1),HEAD(10)
16620      INTEGER LINE(102),BLANK,PLUS
16630      DATA BLANK,PLUS /1H,1H+/
16640      NC1=NC+1
16650      WRITE(6,501)
16660 501    FORMAT(1H1)
16670      N10=(NC-1)/10
16680      XMIN=X(1)
16690      XMAX=X(1)
16700      YMIN=Y(1)
16710      YMAX=Y(1)
16720      DO 6 I=1,N
16730      IF(X(I).LT.XMIN) XMIN=X(I)
16740      IF(X(I).GT.XMAX) XMAX=X(I)
16750      IF(Y(I).LT.YMIN) YMIN=Y(I)
16760      IF(Y(I).GT.YMAX) YMAX=Y(I)
16770 6      CONTINUE
16780      DELX=(XMAX-XMIN)/(NC-1)
16790      DELY=(YMAX-YMIN)/(NR-1)
16800      DO 5 I=1,N10
16810 5      HEAD(I)=XMIN+10*I*DELX
16820      DO 11 K=1,NC1
16830      LINE(K)='- '
16840      IF(MOD(K,5).EQ.1) LINE(K)='.'
16850 11     CONTINUE
16860      WRITE(6,504) (LINE(K),K=1,NC1)
16870      YU=YMAX+DELY/2.
16880      YL=YMAX-DELY/2.
16890      DO 7 J=1,NR
16900      DO 8 K=1,NC
16910 8      LINE(K)=BLANK
16920      LINE(NC1)='I'
16930      DO 9 I=1,N
16940      IF(Y(I).GE.YU) GO TO 9
16950      IF(Y(I).LT.YL) GO TO 9
16960      K=(X(I)-XMIN)/DELX+1.5
16970      IF(LINE(K).NE.BLANK) LINE(K)=PLUS
16980      IF(LINE(K).EQ.BLANK) LINE(K)=ISYM(I)
16990 9      CONTINUE
17000      YM=(YU+YL)/2.
17010      WRITE(6,508) YM,(LINE(K),K=1,NC1)
17020      YU=YL
17030      YL=YL-DELY
17040 7      CONTINUE
17050      DO 12 K=1,NC1
17060      LINE(K)='- '
17070      IF(MOD(K,5).EQ.1) LINE(K)='.'
17080 12     CONTINUE
17090      WRITE(6,504) (LINE(K),K=1,NC1)

```

```

17160      DO 13 K=1,NC1
17110      LINE(K)=BLANK
17120      IF(MOD(K,10).EQ.1) LINE(K)='I'
17130 13    CONTINUE
17140      WRITE( 6,3002) LY,LX,(LINE(K),K=1,NC1)
17150      WRITE( 6,507) XMIN,(HEAD(I),I=1,N10)
17160      WRITE(6,502) N
17170 502   FORMAT(/,2X,I4,15H POINTS PLOTTED)
17180      RETURN
17190 504   FORMAT(1X,14(1H-),102A1)
17200 507   FORMAT(10X,11(F10.4))
17210 508   FORMAT(1X,F12.4,1X,1HI,102A1)
17220 3002  FORMAT(4X,A1,5H / ,A1,4X,102A1)
17230      END
17240/END

```

PRECEDING PAGE

```

00100BOTS,P30,T100,CH140000,STCX3.
00110ACCOUNT,NO3265,BOT2A.
00120BANNERS(OUTPUT)*J. PUTNAM*DEPT 220*BLD 110-4**
00130ATTACH(INFIL=YSSX)
00140FTN(R=0,DPT=1)
00164LDSET(PRESET=INDEF)
00170LGO.
00190/EOR
00200      PROGRAM BOTSCM(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
00210      1 INFIL,TAPE1=INFIL)
00220C
00230C BOTSCM IS THE BOT MONO-STATIC SCATTERING CODE.
00240C
00250C UNIT 5 IS THE CARD READER.
00260C UNIT 6 IS THE LINE PRINTER.
00270C UNIT 1 IS A DISK FILE CONTAINING THE Y MATRIX.
00280C
00285      COMPLEX U
00300      COMPLEX STT(91),SPP(91),STP(91),SPT(91)
00310      COMPLEX CBT(82),CBP(82),CWT(59),CWP(59),CCT(96),CCP(96)
00320      COMPLEX Y(2304)
00325      COMPLEX RBT(736),RBP(736),RWT(368),RWP(368)
00326      COMPLEX RCT(2208),RCP(2208)
00327      COMMON /WAVE/ BK
00330      COMMON /BOT1/ NMODE,NPT,NBAND
00340      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
00350      COMMON /BOT3/ OH(82),SV(82),CV(82)
00360      COMMON /BOT5/ T(160),TP(160),TZ(160)
00370      COMMON /BOT6/ IEDGE,IUNIF
00382      COMMON /WIRE1/ NPH,XW(101),YW(101),ZW(101),
00384      1 XW1(100),YW1(100),ZW1(100)
00385      COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
00386      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
00387      COMMON /CAP1/ NC,XC,YC,ZC(2)
00388      COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
00389      COMMON /EDG1/ NE,ZE(2),ZBE(10)
00400      DIMENSION THS(73),PHIS(73)
00410      DIMENSION ANG(6),IPLANE(6),ANG1(6),ANG2(6)
00420      DIMENSION MORD(2)
00430      U=(0,1.)
00440      ETA=376.707
00450      PI=3.14159265
00460      DTOR=PI/180.
00462      READ(5,1) BK
00463      1 FORMAT(E15.7)
00464      WRITE(6,2) BK
00465      2 FORMAT(9H1      BK,/,E15.7)
00470C
00480      CALL BOTIN
00490C
00500      CALL CAPIN
00510C

```

```

00520      CALL WIREIN
00522C
00530C READ INPUTS, DESCRIBING THE SCATTERING PLANES.
00540      READ(5,44) NANG,NT,PHI1,THI
00550 44    FORMAT(2I3,2F8.4)
00560      READ(5,53) (ANG(I),I=1,NANG)
00590 53    FORMAT(10F8.4)
00610C IPLANE = 1 FOR PHI FIXED,
00620C          2 FOR THETA FIXED.
00630      READ(5,54) (IPLANE(I),I=1,NANG)
00640 54    FORMAT(10I8)
00642      READ(5,53) (ANG1(I),I=1,NANG)
00644      READ(5,53) (ANG2(I),I=1,NANG)
00650      WRITE(6,3) NANG,NT
00660 3    FORMAT(//,25H NUMBER OF FIXED ANGLES =,I3,//
00670      1 35H NUMBER OF ANGLES PER FIXED ANGLE =,I4)
00680      WRITE(6,4) (ANG(I),IPLANE(I),ANG1(I),ANG2(I),I=1,NANG)
00690 4    FORMAT(//,45H FIXED ANGLE CODE VARIABLE ANGLE RANGE,
00700      1 //,(F8.1,9X,I2,6X,F8.1,3H - ,F8.1))
00710      LS=NP-3
00712      NM=LS/2
00714      NM1=NM+1
00716      NM4=NM*4
00720      LSS=LS+LS
00722      LR=(NPR-3)/2
00724      LC=NC*NM*LR
00726      LC2=LC*2
00728      LE=NE*NM
00730      LSC=LS*(LC2+LE)
00740      LSM=LS*(LW+NJ)
00750      LCC=(LC2+LE)**2
00760      LCM=(LC2+LE)*(LW+NJ)
00770      LW=(LW+NJ)**2
00810      BKL=BK*BL
00811      P1=(BK*BK*ETA/2./PI)**2/4./PI
00812      KMODE=2*NMODE-1
00813      CALL ORDER(MORD)
00814      DO 90 IANG=1,NANG
00815      REWIND 1
00816      READ(1) I
00817      READ(1) I
00818C INITIALIZATION ARRAYS, AND SET THE SCATTERING ANGLES.
00819      IPLAN=IPLANE(IANG)
00840      DT=(ANG2(IANG)-ANG1(IANG))/(NT-1)
00883      DO 100 K=1,NT
00886      PHIS(K)=ANG(IANG)*(2-IPLAN)+(ANG1(IANG)+DT*(K-1))*(IPLAN-1)
00887      THS(K)=ANG(IANG)*(IPLAN-1)+(ANG1(IANG)+DT*(K-1))*(2-IPLAN)
00888      STT(K)=0.0
00889      SPP(K)=0.0
00890      STP(K)=0.0
00891 100    SPT(K)=0.0
00892      CALL RROT(RBT,RBP,NT,THS,PHIS)

```

```

00894      IF(NC.NE.0) CALL RCAP(RCT,RCP,NT,THS,PHIS)
00896      IF(NPW.NE.0) CALL RWIRE(RWT,RWP,NT,THS,PHIS)
00900      NUMB=0
00910  50    CONTINUE
00920      READ(1) M,N
00930      IF(EOF(1)) 37,60
00931  60    NUMB=NUMB+1
00932      DO 120 K=1,NT
00933          KB=(K-1)*LS
00934          KC=(K-1)*(LC2+LE)
00935          KM=(K-1)*(LW+NJ)
00940          DMT=2.*BL*SINC(M*PI+BKL*COS(THS(K)*DTOR))
00942          DNZ=DMT-2.*BL*(-1)**M*SINC(BKL*COS(THS(K)*DTOR))
00950C DETERMINE SUBMATRIX TYPE, AND CALCULATE CURRENTS.
00960      IF(M.LT.NMODE) GO TO 61
00970      IF(M.EQ.MORD(1)) GO TO 62
00980      IF(M.EQ.MORD(2)) GO TO 63
00990      GO TO 120
00992  61    DMT=2.*BL*SINC(-M*PI+BKL*COS(THS(K)*DTOR))
00994          DNZ=DMT-2.*BL*(-1)**M*SINC(BKL*COS(THS(K)*DTOR))
01010      IF(M.LT.NMODE) GO TO 71
01020      IF(M.EQ.MORD(1)) GO TO 72
01030      IF(M.EQ.MORD(2)) GO TO 73
01040      GO TO 120
01050  62    IF(M.LT.NMODE) GO TO 74
01060      IF(M.EQ.MORD(1)) GO TO 75
01070      IF(M.EQ.MORD(2)) GO TO 76
01080      GO TO 120
01090  63    IF(M.LT.NMODE) GO TO 77
01100      IF(M.EQ.MORD(1)) GO TO 78
01110      IF(M.EQ.MORD(2)) GO TO 79
01120      GO TO 120
01130C BOT-BOT.
01140  71    IF(K.EQ.1) READ(1) (Y(I),I=1,LSS)
01162      CALL ZERO(LS,CBT)
01164      CALL ZERO(LS,CBP)
01170      CALL MULTYR(LS,LS,Y,DMT,DNZ,RBT(KB+1),CBT)
01175      CALL MULTYR(LS,LS,Y,DMT,DNZ,RBP(KB+1),CBP)
01176      CALL MULTRC(LS,DMT,DMZ,RBT(KB+1),CBT,STT(K))
01177      CALL MULTRC(LS,DMT,DMZ,RBP(KB+1),CBP,SPP(K))
01178      CALL MULTRC(LS,DMT,DMZ,RBT(KB+1),CBP,STP(K))
01179      CALL MULTRC(LS,DMT,DMZ,RBP(KB+1),CBP,SPT(K))
01180      GO TO 120
01190C CAP-BOT.
01200  72    IF(K.EQ.1) READ(1) (Y(I),I=1,LSC)
01202      CALL ZERO(LC2+LE,CCT)
01204      CALL ZERO(LC2+LE,CCP)
01210      CALL MULTYR(LC2+LE,LS,Y,DMT,DNZ,RBT(KB+1),CCT)
01215      CALL MULTYR(LC2+LE,LS,Y,DMT,DNZ,RBP(KB+1),CCP)
01216      CALL MULTRC(LC2+LE,1.0,1.0,RCT(KC+1),CCT,STT(K))
01217      CALL MULTRC(LC2+LE,1.0,1.0,RCP(KC+1),CCP,SPP(K))
01218      CALL MULTRC(LC2+LE,1.0,1.0,RCT(KC+1),CCP,STP(K))

```

```

01219 CALL MULTRC(LC2+LE,1.0,1.0,RCP(KC+1),CCT,SPT(K))
01220 GO TO 120
01230C WIRE-BOT.
01240 73 IF(K.EQ.1) READ(1) (Y(I),I=1,LSW)
01241 CALL ZERO(LW+NJ,CWT)
01242 CALL ZERO(LW+NJ,CWP)
01243 CALL MULTYR(LW+NJ,LS,Y,DNT,DNZ,RBT(KB+1),CWT)
01244 CALL MULTYR(LW+NJ,LS,Y,DNT,DNZ,RBP(KB+1),CWP)
01245 CALL MULTRC(LW+NJ,1.0,1.0,RWT(KW+1),CWT,STT(K))
01246 CALL MULTRC(LW+NJ,1.0,1.0,RWP(KW+1),CWP,SPP(K))
01247 CALL MULTRC(LW+NJ,1.0,1.0,RWT(KW+1),CWP,STP(K))
01248 CALL MULTRC(LW+NJ,1.0,1.0,RWP(KW+1),CWT,SPT(K))
01249 GO TO 120
01250C BOT-CAP.
01260 74 IF(K.EQ.1) READ(1) (Y(I),I=1,LSC)
01261 CALL ZERO(LS,CBT)
01262 CALL ZERO(LS,CBP)
01263 CALL MULTYR(LS,LC2+LE,Y,1.0,1.0,RCT(KC+1),CBT)
01264 CALL MULTYR(LS,LC2+LE,Y,1.0,1.0,RCP(KC+1),CBP)
01265 CALL MULTRC(LS,DMT,DMZ,RBT(KB+1),CBT,STT(K))
01266 CALL MULTRC(LS,DMT,DMZ,RBP(KB+1),CBP,SPP(K))
01267 CALL MULTRC(LS,DMT,DMZ,RBT(KB+1),CBP,STP(K))
01268 CALL MULTRC(LS,DMT,DMZ,RBP(KB+1),CBT,SPT(K))
01269 GO TO 120
01270C CAP-CAP.
01280 75 IF(K.EQ.1) READ(1) (Y(I),I=1,LCC)
01281 CALL ZERO(LC2+LE,CCT)
01282 CALL ZERO(LC2+LE,CCP)
01283 CALL MULTYR(LC2+LE,LC2+LE,Y,1.0,1.0,RCT(KC+1),CCT)
01284 CALL MULTYR(LC2+LE,LC2+LE,Y,1.0,1.0,RCP(KC+1),CCP)
01285 CALL MULTRC(LC2+LE,1.0,1.0,RCT(KC+1),CCT,STT(K))
01286 CALL MULTRC(LC2+LE,1.0,1.0,RCP(KC+1),CCP,SPP(K))
01287 CALL MULTRC(LC2+LE,1.0,1.0,RCT(KC+1),CCP,STP(K))
01288 CALL MULTRC(LC2+LE,1.0,1.0,RCP(KC+1),CCT,SPT(K))
01289 GO TO 120
01290C WIRE-CAP.
01300 76 IF(K.EQ.1) READ(1) (Y(I),I=1,LCW)
01301 CALL ZERO(LW+NJ,CWT)
01302 CALL ZERO(LW+NJ,CWP)
01303 CALL MULTYR(LW+NJ,LC2+LE,Y,1.0,1.0,RCT(KC+1),CWT)
01304 CALL MULTYR(LW+NJ,LC2+LE,Y,1.0,1.0,RCP(KC+1),CWP)
01305 CALL MULTRC(LW+NJ,1.0,1.0,RWT(KW+1),CWT,STT(K))
01306 CALL MULTRC(LW+NJ,1.0,1.0,RWP(KW+1),CWP,SPP(K))
01307 CALL MULTRC(LW+NJ,1.0,1.0,RWT(KW+1),CWP,STP(K))
01308 CALL MULTRC(LW+NJ,1.0,1.0,RWP(KW+1),CWT,SPT(K))
01309 GO TO 120
01310C BOT-WIRE.
01320 77 IF(K.EQ.1) READ(1) (Y(I),I=1,LSW)
01321 CALL ZERO(LS,CBT)
01322 CALL ZERO(LS,CBP)
01323 CALL MULTYR(LS,LW+NJ,Y,1.0,1.0,RWT(KW+1),CBT)
01324 CALL MULTYR(LS,LW+NJ,Y,1.0,1.0,RWP(KW+1),CBP)

```

```

C1416 CALL MULTRC(LS,DMT,DMZ,RBT(KB+1),CBT,STT(K))
C1417 CALL MULTRC(LS,DMT,DMZ,RBP(KB+1),CBP,SPP(K))
C1418 CALL MULTRC(LS,DMT,DMZ,RBT(KB+1),CBP,STP(K))
C1419 CALL MULTRC(LS,DMT,DMZ,RBP(KB+1),CBT,SPT(K))
C1420 GO TO 120
C1430C CAP-WIRE.
C1440 78 IF(K.EQ.1) READ(1) (Y(I),I=1,LCW)
C1442 CALL ZERO(LC2+LE,CCT)
C1444 CALL ZERO(LC2+LE,CCP)
C1450 CALL MULTYR(LC2+LE,LW+NJ,Y,1.0,1.0,RWT(KW+1),CCT)
C1455 CALL MULTYR(LC2+LE,LW+NJ,Y,1.0,1.0,RWP(KW+1),CCP)
C1456 CALL MULTRC(LC2+LE,1.0,1.0,RCT(KC+1),CCT,STT(K))
C1457 CALL MULTRC(LC2+LE,1.0,1.0,RCP(KC+1),CCP,SPP(K))
C1458 CALL MULTRC(LC2+LE,1.0,1.0,RCT(KC+1),CCP,STP(K))
C1459 CALL MULTRC(LC2+LE,1.0,1.0,RCP(KC+1),CCT,SPT(K))
C1460 GO TO 120
C1470C WIRE-WIRE.
C1480 79 IF(K.EQ.1) READ(1) (Y(I),I=1,LW)
C1482 CALL ZERO(LW+NJ,CWT)
C1484 CALL ZERO(LW+NJ,CWP)
C1490 CALL MULTYR(LW+NJ,LW+NJ,Y,1.0,1.0,RWT(KW+1),CWT)
C1495 CALL MULTYR(LW+NJ,LW+NJ,Y,1.0,1.0,RWP(KW+1),CWP)
C1496 CALL MULTRC(LW+NJ,1.0,1.0,RWT(KW+1),CWT,STT(K))
C1497 CALL MULTRC(LW+NJ,1.0,1.0,RWP(KW+1),CWP,SPP(K))
C1498 CALL MULTRC(LW+NJ,1.0,1.0,RWT(KW+1),CWP,STP(K))
C1499 CALL MULTRC(LW+NJ,1.0,1.0,RWP(KW+1),CWT,SPT(K))
C1500 120 CONTINUE
C1510 GO TO 50
C1550C ALL Y SUBMATRICES HAVE BEEN READ.
C1560 37 CONTINUE
C1562 WRITE(6,114) NUMB
C1564 114 FORMAT(//,1X,I4,17H SUBMATRICES READ)
C1630C *** END CALCULATION OF RCS ***
C1725C PRINT OUT RCS.
C1730 WRITE(6,403)
C1735 403 FORMAT(1H1,32X,19HMONO-STATIC RCS(DB),/,2CX,45(1H-),/,
C1740 1 55H PHI THETA 00 00 00
C1745 2 8H 00,/,
C1750 3 55H+ -- // -/
C1755 4 8H /-/,)
C1780 DO 110 K=1,NT
C1939 X2=CABS(STT(K))
C1940 X2=DBCON(P1+X2+X2)
C1945 X3=CABS(SPP(K))
C1950 X3=DBCON(P1+X3+X3)
C1955 X4=CABS(STP(K))
C1960 X4=DBCON(P1+X4+X4)
C1965 X5=CABS(SPT(K))
C1970 X5=DBCON(P1+X5+X5)
C1975 WRITE(6,113) PHIS(K),THS(K),X2,X3,X4,X5
C1980 113 FORMAT(1X,2F8.1,4F12.2)
C1990 110 CONTINUE

```

```

02600 90  CONTINUE
02640      STOP
02645      END
-----
02821C  SUBROUTINE ZERO(N,A)
02822      A=0.
02823C  COMPLEX A(1)
02824      DO 100 I=1,N
02825      A(I)=C.0
02826      100 CONTINUE
02827      RETURN
02828      END
-----
02830C  SUBROUTINE MULTYR(NI,NJ,Y,DT,DZ,V,C)
02840      UPDATE CURRENT ARRAY C BY Y * (DT,DZ) * V.
02850C  COMPLEX S,Y(1),V(1),C(1)
02855      NJ2=NJ/2
02890      DO 100 I=1,NI
02900      S=0.C
02910      IND=I
02920      DO 50 J=1,NJ
02922      D=DT
02924      IF(J.GT.NJ2) D=DZ
02930      S=S+Y(IND)*D*V(J)
02940  50  IND=IND+NI
02950      C(I)=C(I)+S
02960  100 CONTINUE
02970      RETURN
02980      END
-----
03100C  SUBROUTINE MULTRC(NI,DT,DZ,R,C,G)
03110      UPDATE G BY (DT,DZ) * R * C.
03120C  COMPLEX G,R(1),C(1)
03150      NI2=NI/2
03160      DO 100 I=1,NI
03162      D=DT
03164      IF(I.GT.NI2) D=DZ
03170      G=G+D*R(I)*C(I)
03180  100 CONTINUE
03190      RETURN
03200      END
-----
03210C  *****
03220  FUNCTION SINC(X)
03230C  *****
03240      SINC=1.C
03250      IF(X.NE.0.0) SINC=SIN(X)/X
03260      RETURN
03270      END

```

```

03280C *****
03290C FUNCTION DBCON(X)
03300C *****
03310C RETURN X IN DB,
03320C DBCON=-1000.0
03330C IF(X.EQ.0.0) RETURN
03340C DBCON=10.*ALOG10(X)
03350C RETURN
03360C END
03370C -----
03375C SUBROUTINE BOTIN
03380C
03385C READ BOT COORDINATES AND COMPUTE BOT SEGMENT ARRAYS.
03390C
03395C COMMON /BOT1/ NMODE,NPT,NBAND
03400C COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
03405C COMMON /BOT3/ DH(82),SV(82),CV(82)
03410C COMMON /BOT5/ T(160),TP(160),TZ(160)
03415C COMMON /BOT6/ IEDGE,IUNIF
03420C READ(5,49) NMODE,NPT,NBAND
03425C READ(5,49) NP
03430C 49 FORMAT(3I3)
03435C WRITE(6,48) NMODE,NPT,NBAND,NP
03440C 48 FORMAT(32H NMODE NPT NBAND NP,/,4I8)
03445C READ(5,53)(YB(I),I=1,NP)
03450C READ(5,53)(XB(I),I=1,NP)
03455C 53 FORMAT(10F8.4)
03460C WRITE(6,55)
03465C 55 FORMAT(/,3H YB)
03470C WRITE(6,46)(YB(I),I=1,NP)
03475C 46 FORMAT(1X,10F8.4)
03480C WRITE(6,56)
03485C 56 FORMAT(/,3H XB)
03490C WRITE(6,46)(XB(I),I=1,NP)
03495C PLOT THE BODY COORDINATES,
03500C CALL PLOTB(XB,YB,NP,41)
03505C READ(5,53) BL
03510C WRITE(6,47) BL
03515C 47 FORMAT(/,21H HALF-LENGTH OF BOT =,F12.4)
03520C CHECK IF BOT CONTAINS AN ODD NUMBER OF POINTS.
03525C IF(MOD(NP,2).NE.1) GO TO 980
03530C CHECK IF GENERATING CURVE IS OPEN OR CLOSED.
03535C IEDGE=1
03540C IF((YB(1)-YB(NP)).NE.0..OR.(XB(1)-XB(NP)).NE.0.) GO TO 10
03545C IEDGE=0
03550C YB(NP+1)=YB(2)
03555C XB(NP+1)=XB(2)
03560C YB(NP+2)=YB(3)
03565C XB(NP+2)=XB(3)
03570C NP=NP+2
03575C WRITE(6,66) NP
03580C 66 FORMAT(/,39H BOT GENERATING CURVE IS CLOSED. NP = ,I3)

```

AD-A118 498

MCDONNELL DOUGLAS RESEARCH LABS ST LOUIS MO
ALGORITHM FOR SURFACE OF TRANSLATION ATTACHED RADIATORS (A-STAR--ETC(1))
MAY 82 L N MEDGYESI-MITSCHANG, J M PUTNAM F30602-80-C-0106
RADC-TR-82-113-VOL-3 NL

UNCLASSIFIED

3 of 3
AD-A118 498
16-4-86

END
DATE
FILMED
9 82
DTIC

```

03585 10 CONTINUE
03590C COMPUTATION OF BODY SEGMENT PARAMETERS.
03595 DO 57 I=2,NP
03600 I2=I-1
03605 RR1=YB(I)-YB(I2)
03610 RR2=XB(I)-XB(I2)
03615 DH(I2)=SQRT(RR1*RR1+RR2*RR2)
03620 XB1(I2)=.5*(XB(I)+XB(I2))
03625 YB1(I2)=.5*(YB(I)+YB(I2))
03630 SV(I2)=RR1/DH(I2)
03635 CV(I2)=RR2/DH(I2)
03640 57 CONTINUE
03645C CHECK IF BOT SEGMENTATION IS UNIFORM.
03650 IUNIF=0
03655 NP1=NP-1
03660 DO 60 I=2,NP1
03665 RR1=DH(I)/DH(1)
03670 IF(RR1.LT.0.99 .OR. RR1.GT.1.01) GO TO 20
03675 60 CONTINUE
03680 IUNIF=1
03685 WRITE(6,67)
03690 67 FORMAT(/, ' BOT GENERATING CURVE HAS UNIFORM SEGMENTATION')
03695 20 CONTINUE
03700C COMPUTATION OF TRIANGLE FUNCTIONS T.
03705 NM=(NP-3)/2
03710 DO 74 J=1,NM
03715 J2=2*(J-1)+1
03720 J3=J2+1
03725 J4=J3+1
03730 J5=J4+1
03735 J6=4*(J-1)+1
03740 J7=J6+1
03745 J8=J7+1
03750 J9=J8+1
03755 DEL1=DH(J2)+DH(J3)
03760 DEL2=DH(J4)+DH(J5)
03765 TP(J6)=1./DEL1
03770 TP(J7)=1./DEL1
03775 TP(J8)=-1./DEL2
03780 TP(J9)=-1./DEL2
03785 T(J6)=DH(J2)/2./DEL1
03790 T(J7)=(DH(J2)+DH(J3)/2.)/DEL1
03795 T(J8)=(DH(J4)/2.+DH(J5))/DEL2
03800 T(J9)=DH(J5)/2./DEL2
03805 74 CONTINUE
03810 NM4=NM*4
03815 DO 75 J=1,NM4
03820 75 TZ(J)=T(J)
03825 IF(IEDGE.EQ.0) GO TO 76
03830 TZ(1)=2.-T(1)
03835 TZ(2)=2.-T(2)
03840 TZ(NM4-1)=2.-T(NM4-1)

```

```

03845      TZ(NM4)=2.-T(NM4)
03850 76    CONTINUE
03855      RETURN
03860 980    WRITE(6,981)
03865 981    FORMAT(/, ' **** EPROR IN BOT INPUT ')
03870      STOP
03875      END
04410C *****
04420      SUBROUTINE PLOTB(X,Y,N,NR)
04430C *****
04440C
04450C WRITTEN 2/14/74   BY J. M. PUTNAM           DEPT 220       X23877
04460C
04470C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
04480C N IS THE NUMBER OF POINTS TO BE PLOTTED.
04490C NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
04500C NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
04510C NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
04520C
04530      REAL X(1),Y(1),HEAD(10)
04540      INTEGER LINE(101),BLANK,STAR,PLUS
04550      DATA BLANK,STAR,PLUS /1H,1H*,1H+/
04560      NC=51
04570      N10=(NC-1)/10
04580      WRITE(6,500)
04590 500    FORMAT(/,44H BODY COORDINATES, + INDICATES TRIANGLE PEAK)
04600      WRITE(6,504)
04610      XMIN=X(1)
04620      XMAX=X(1)
04630      YMIN=Y(1)
04640      YMAX=Y(1)
04650      DO 6 I=1,N
04660      IF(X(I).LT.XMIN) XMIN=X(I)
04670      IF(X(I).GT.XMAX) XMAX=X(I)
04680      IF(Y(I).LT.YMIN) YMIN=Y(I)
04690      IF(Y(I).GT.YMAX) YMAX=Y(I)
04700 6      CONTINUE
04710      DEL=XMAX-XMIN
04720      IF(YMAX-YMIN.GT.DEL) DEL=YMAX-YMIN
04730      XMAX=XMIN+DEL
04740      YMAX=YMIN+DEL
04750      DO 5 I=1,N10
04760      Z=I
04770 5      HEAD(I)=(XMAX-XMIN)*Z/N10+XMIN
04780      DY=(YMAX-YMIN)/(NR-1)
04790      IEDGE=1
04800      IF(X(1).EQ.X(N) .AND. Y(1).EQ.Y(N)) IEDGE=0
04810      Z=YMAX+DY
04820      YL=Z-DY/2.
04830      DO 7 J=1,NR
04840      DO 8 K=1,NC
04850 8      LINE(K)=BLANK

```

```

04860      Z=Z-DY
04870      YU=YL
04880      YL=Z-DY/2.
04890      DO 9 I=1,N
04900      IF(Y(I).GE.YU) GO TO 9
04910      IF(Y(I).LT.YL) GO TO 9
04920      K=(X(I)-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
04930      IF(K.GT.NC) K=NC
04940      LINE(K)=STAR
04950      IF(MOD(I,2).EQ.1) LINE(K)=PLUS
04960      IF(IEDGE.EQ.C) GO TO 9
04970      IF(I.EQ.1 .OR. I.EQ.N) LINE(K)=STAR
04980 9      CONTINUE
04990      WRITE(6,508) Z,(LINE(K),K=1,NC)
05000 7      CONTINUE
05010      WRITE(6,504)
05020      WRITE(6,3002)
05030      WRITE(6,507) XMIN,(HEAD(I),I=1,N10)
05040      RETURN
05050 504  FORMAT (1X,14(1H-),1H.,10(5H----.),1H- )
05060 507  FORMAT(10X,11(F10.4))
05070 508  FORMAT(1X,F12.4,1X,1HI,51A1,1HI )
05080 3002 FORMAT(4X,7HYH / XH,4X,1HI,5(9X,1HI))
05090      END
05100C -----
05102C      SUBROUTINE CAPIN
05104C
05106C      READ CAP INPUTS AND COMPUTE CAP ARRAYS.
05108C
05110      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
05112      COMMON /CAP1/ NC,XC,YC,ZC(2)
05114      COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
05116      COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
05118      COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
05120      COMMON /EDG1/ NE,ZE(2),ZBE(10)
05122      COMMON /EDG2/ TCE(10),TPCE(10),TBE(10),TPBE(10)
05124      READ(5,1) NC,NPR,NE
05126 1      FORMAT(3I3)
05127      IF(NE.NE.0) NE=NC
05128      WRITE(6,3) NC,NPR,NE
05130 3      FORMAT(16H1 NC NPR NE,/,3I5,/)
05132      IF(NC.EQ.0) RETURN
05134      READ(5,2) XC,YC
05136 2      FORMAT(10F8.4)
05138      READ(5,2) (ZC(I),I=1,NC)
05142      READ(5,2) (RHOC(I),I=1,NPR)
05143      IF(NE.NE.0) READ(5,2) (ZE(I),I=1,NE)
05144      WRITE(6,4)
05146 4      FORMAT(37H CAP XC YC ZC ZE,/)
05148      DO 100 I=1,NC
05149      IF(NE.EQ.0) WRITE(6,5) I,XC,YC,ZC(I)
05150      IF(NE.NE.0) WRITE(6,5) I,XC,YC,ZC(I),ZE(I)

```

```

C5151 100 CONTINUE
C5152 5 FORMAT(I4,4X,4F8.4)
C5154 WRITE(6,6)
C5156 6 FORMAT(/,5H RHOC)
C5158 WRITE(6,7) (RHOC(I),I=1,NPR)
C5160 7 FORMAT(1X,10F8.4)
C5162 IF(MOD(NPR,2).NE.1) GO TO 990
C5164 DO 120 I=2,NPR
C5166 120 IF(RHOC(I).LE.RHOC(I-1)) GO TO 980
C5168 C COMPUTATION OF CAP SECTOR PARAMETERS.
C5170 DO 47 I=1,NP
C5172 RC(I)=SQRT((YB(I)-YC)**2+(XB(I)-XC)**2)
C5174 47 CONTINUE
C5176 DO 57 I=2,NP
C5178 I2=I-1
C5180 RR1=YB(I2)-YC
C5182 RR2=XB(I2)-XC
C5184 RC1(I2)=SQRT(RR1*RR1+RR2*RR2)
C5186 AC(I2)=ABS((XB(I)-XC)*(YB(I)+YC)+(XB(I2)-XB(I))*
C5188 1 (YB(I2)+YB(I))+(XC-XB(I2))*(YC+YB(I2)))/2.
C5190 SPC(I2)=RR1/RC1(I2)
C5192 CPC(I2)=RR2/RC1(I2)
C5194 57 CONTINUE
C5196 DO 67 I=2,NPR
C5198 I2=I-1
C5200 RHOC(I2)=(RHOC(I)+RHOC(I2))/2.
C5202 DRHOC(I2)=RHOC(I)-RHOC(I2)
C5204 67 CONTINUE
C5206 C COMPUTATION OF CAP TRIANGLE FUNCTIONS.
C5208 LC=(NPR-3)/2
C5210 DO 74 J=1,LC
C5212 J2=2*(J-1)+1
C5214 J3=J2+1
C5216 J4=J3+1
C5218 J5=J4+1
C5220 J6=4*(J-1)+1
C5222 J7=J6+1
C5224 J8=J7+1
C5226 J9=J8+1
C5228 DEL1=DRHOC(J2)+DRHOC(J3)
C5230 DEL2=DRHOC(J4)+DRHOC(J5)
C5232 TPCR(J6)=1./DEL1
C5234 TPCR(J7)=1./DEL1
C5236 TPCR(J8)=-1./DEL2
C5238 TPCR(J9)=-1./DEL2
C5240 TCR(J6)=DRHOC(J2)/2./DEL1
C5242 TCR(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
C5244 TCR(J8)=(DRHOC(J4)/2.+DRHOC(J5))/DEL2
C5246 TCR(J9)=DRHOC(J5)/2./DEL2
C5248 74 CONTINUE
C5250 LC4=LC*4
C5252 DO 75 I=1,LC4

```

```

05254      TCT(I)=TCR(I)
05256      TPCT(I)=TPCR(I)
05258 75      CONTINUE
05260      TCT(LC4-1)=2.-TCT(LC4-1)
05262      TCT(LC4)=2.-TCT(LC4)
05264      TPCT(LC4-1)=-TPCT(LC4-1)
05266      TPCT(LC4)=-TPCT(LC4)
05268      IF(RHOC(1).EQ.0.0) GO TO 76
05270      TCT(1)=2.-TCT(1)
05272      TCT(2)=2.-TCT(2)
05274      TPCT(1)=-TPCT(1)
05276      TPCT(2)=-TPCT(2)
05278 76      CONTINUE
05279      IF(NE.EQ.0) RETURN
05280C COMPUTATION OF EDGE HALF TRIANGLE FUNCTIONS.
05282      DD 80 IC=1,NC
05284      J2=NPR-2
05286      J3=J2+1
05288      J6=2+(IC-1)+1
05290      J7=J6+1
05292      DEL1=DRHOC(J2)+DRHOC(J3)
05294      TPCE(J6)=1./DEL1
05296      TPCE(J7)=1./DEL1
05298      TCE(J6)=DRHOC(J2)/2./DEL1
05300      TCE(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
05302      DEL2=ZE(IC)-ZC(IC)
05304      ZBE(J6)=ZC(IC)+0.25*DEL2
05306      ZBE(J7)=ZC(IC)+0.75*DEL2
05308      IF(DEL2.LT.0.0) GO TO 78
05310C EDGE IS AT Z=-L.
05312      TPBE(J6)=-1./DEL2
05314      TPBE(J7)=-1./DEL2
05316      TBE(J6)=-0.75
05318      TBE(J7)=-0.25
05320      GO TO 80
05322 78      CONTINUE
05324C EDGE IS AT Z=+L.
05326      TPBE(J6)=1./DEL2
05328      TPBE(J7)=1./DEL2
05330      TBE(J6)=-0.75
05332      TBE(J7)=-0.25
05334 80      CONTINUE
05336      RETURN
05338 980      WRITE(6,981)
05340 981      FORMAT(//,1) **** ERROR IN CAP INPUT!
05342      STOP
05344      END
-----
05350C SUBROUTINE WIREIN
05352C READ WIRE COORDINATES AND COMPUTE WIRE SEGMENT ARRAYS.
05353C

```

```

05535 COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
05540 1 XW1(100),YW1(100),ZW1(100)
05545 COMMON /WIRE2/ DHW(100),DXW(100),DYW(100),DZW(100)
05550 COMMON /WIRE3/ NW,INDW(6),RADW(5)
05555 COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
05560 COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RAOD(10)
05565 COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
05570 COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
05575 COMMON /JUNC4/ UXJ(10),UYJ(10),UZJ(10)
05580 COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
05585 COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
05590 NW=0
05595 NJ=0
05600 LW=0
05605 READ(5,51) NW,NPW,NJ
05610 51 FORMAT(3I3)
05615 WRITE(6,1) NW,NPW,NJ
05620 1 FORMAT(24H1 NW NPW NJ,/,3I6)
05625 IF(NW.EQ.0) RETURN
05630 READ(5,53) (XW(I),I=1,NPW)
05635 READ(5,53) (YW(I),I=1,NPW)
05640 READ(5,53) (ZW(I),I=1,NPW)
05645 53 FORMAT(10F8.4)
05650 READ(5,52) (INDW(I),I=1,NW)
05655 52 FORMAT(10I8)
05660 INDW(NW+1)=NPW+1
05665 READ(5,53) (RADW(I),I=1,NW)
05670 IF(NJ.EQ.0) GO TO 50
05675 C NOTE, INDJW MUST BE MONOTONIC INCREASING.
05680 READ(5,52) (INDJW(I),I=1,NJ)
05685 READ(5,53) (RAOD(I),I=1,NJ)
05690 READ(5,53) (UXJ(I),I=1,NJ)
05695 READ(5,53) (UYJ(I),I=1,NJ)
05700 READ(5,53) (UZJ(I),I=1,NJ)
05705 50 IERW=0
05710 IERJ=0
05715 WRITE(6,61)
05720 61 FORMAT(1,22X,'WIRE COORDINATES',20X,'JUNCTION PARAMETERS',/,
05725 1 26X,'IW',4X,'XW',6X,'YW',6X,'ZW',
05730 2 10X,'IJ',4X,'RADW',5X,'UXJ',5X,'UYJ',5X,'UZJ')
05735 IJ=1
05740 C THIS LOOP LISTS WIRE/JUNCTION POINTS, WHILE CHECKING THE FOLLOWING:
05745 C 1) EACH WIRE MUST CONTAIN AN ODD NUMBER OF POINTS.
05750 C 2) EACH JUNCTION MUST EITHER START OR TERMINATE A WIRE.
05755 C 3) CHECK THAT ALL JUNCTION POINTS ARE FOUND.
05760 DO 100 IW=1,NW
05765 WRITE(6,62) IW,RADW(IW)
05770 62 FORMAT(2X,'WIRE',I3,' RADW=',F8.4)
05775 I1=INDW(IW)
05780 I2=INDW(IW+1)-1
05785 C CHECK FOR AN ODD NUMBER OF POINTS ON WIRE IW.
05790 IF(MOD(I2-I1+1,2).NE.1) IERW=1

```

```

05795      DO 90 I=I1,I2
05800      WRITE(6,63) I,XW(I),YW(I),ZW(I)
05805 63    FORMAT(25X,I3,3F8.4)
05810C CHECK IF WIRE POINT I IS A JUNCTION POINT.
05815      IF(IJ.GT.NJ) GO TO 90
05820      IF(INDJW(IJ).NE.I) GO TO 90
05825C CHECK THAT JUNCTION POINT IJ IS AT THE START OR END OF WIRE.
05830      IF(I.NE.I1.AND.I.NE.I2) IERJ=1
05835      WRITE(6,64) IJ,RADD(IJ),UXJ(IJ),UYJ(IJ),UZJ(IJ)
05840 64    FORMAT(1+1,58X,I3,4F8.4)
05845C DETERMINE DIRECTION IN WHICH WIRE LEAVES JUNCTION POINT.
05850      INDJ(IJ)=1
05855      IF(I.EQ.I2) INDJ(IJ)=-1
05860C COMPUTE JUNCTION PARAMETERS.
05865      RADJ(IJ)=RADW(IW)
05870      XJ(IJ)=XW(I)
05875      YJ(IJ)=YW(I)
05880      ZJ(IJ)=ZW(I)
05885      IJ=IJ+1
05890 90    CONTINUE
05895 100   CONTINUE
05900C CHECK FOR WIRE OR JUNCTION INPUT ERRORS.
05905      IF(IERW.NE.0) GO TO 980
05910      IF(IJ-1.NE.NJ.OR.IERJ.NE.0) GO TO 990
05915C COMPUTATION OF WIRE SEGMENT PARAMETERS.
05920      DO 57 I=2,NPW
05925      I2=I-1
05930      DXW(I2)=XW(I)-XW(I2)
05935      DYW(I2)=YW(I)-YW(I2)
05940      DZW(I2)=ZW(I)-ZW(I2)
05945      OHW(I2)=SQRT(DXW(I2)**2+DYW(I2)**2+DZW(I2)**2)
05950      XW1(I2)=0.5*(XW(I)+XW(I2))
05955      YW1(I2)=0.5*(YW(I)+YW(I2))
05960      ZW1(I2)=0.5*(ZW(I)+ZW(I2))
05965 57    CONTINUE
05970C COMPUTATION OF WIRE TRIANGLE FUNCTIONS TW.
05975      LW=0
05980      DO 75 IW=1,NW
05985      I1=INDW(IW)
05990      I2=INDW(IW+1)-1
05995      LW1=(I2-I1-2)/2
06000      DO 74 J=1,LW1
06005      LW=LW+1
06010      J2=2*(J-1)+I1
06015      J3=J2+1
06020      J4=J3+1
06025      J5=J4+1
06030      J6=4*(LW-1)+1
06035      J7=J6+1
06040      J8=J7+1
06045      J9=J8+1
06050      INDTW(LW)=J2

```

```

06055      DEL1=DHW(J2)+DHW(J3)
06060      DEL2=DHW(J4)+DHW(J5)
06065      TPW(J6)=1./DEL1
06070      TPW(J7)=1./DEL1
06075      TPW(J8)=-1./DEL2
06080      TPW(J9)=-1./DEL2
06085      TW(J6)=DHW(J2)/2./DEL1
06090      TW(J7)=(DHW(J2)+DHW(J3)/2.)/DEL1
06095      TW(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
06100      TW(J9)=DHW(J5)/2./DEL2
06105      74 CONTINUE
06110      75 CONTINUE
06115C     NOTE, LW=(INPW-NW)/2-NW
06120      IF(NJ.EQ.0) RETURN
06125C     COMPUTATION OF JUNCTION HALF TRIANGLE FUNCTIONS TJ.
06130      DO 85 IJ=1,NJ
06135      IF(INDJ(IJ).GT.0) GO TO 80
06140C     JUNCTION IS AT THE END OF A WIRE.
06145      J2=INDJW(IJ)-2
06150      J3=J2+1
06155      J6=2*(IJ-1)+1
06160      J7=J6+1
06165      INDJ(IJ)=J2
06170      DEL1=-DHW(J2)-DHW(J3)
06175      TPJ(J6)=1./DEL1
06180      TPJ(J7)=1./DEL1
06185      TJ(J6)=DHW(J2)/2./DEL1
06190      TJ(J7)=(DHW(J2)+DHW(J3)/2.)/DEL1
06195      GO TO 85
06200C     JUNCTION IS AT THE START OF A WIRE.
06205      80 J4=INDJW(IJ)
06210      J5=J4+1
06215      J8=2*(IJ-1)+1
06220      J9=J8+1
06225      INDJ(IJ)=J4
06230      DEL2=DHW(J4)+DHW(J5)
06235      TPJ(J8)=-1./DEL2
06240      TPJ(J9)=-1./DEL2
06245      TJ(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
06250      TJ(J9)=DHW(J5)/2./DEL2
06255      85 CONTINUE
06260C     COMPUTE UNIT VECTORS FOR JUNCTION(DISK).
06265      DO 95 IJ=1,NJ
06270C     NORMAL UNIT VECTOR (UJ).
06275      RR=SQRT(UXJ(IJ)**2+UYJ(IJ)**2+UZJ(IJ)**2)
06280      UXJ(IJ)=UXJ(IJ)/RR
06285      UYJ(IJ)=UYJ(IJ)/RR
06290      UZJ(IJ)=UZJ(IJ)/RR
06295C     FIND 2 ORTHOGONAL UNIT VECTORS IN THE PLANE OF THE DISK (UJ1 & UJ2).
06300      UXJ1(IJ)=0.0
06305      UYJ1(IJ)=0.0
06310      UZJ1(IJ)=0.0

```

```

06315      UXJ2(IJ)=0.0
06320      UYJ2(IJ)=0.0
06325      UZJ2(IJ)=0.0
06330      IF(UXJ(IJ).EQ.0.0) GO TO 91
06335C FIND INTERSECTION WITH X-Y PLANE.
06340      UYJ1(IJ)=1.0
06345      UXJ1(IJ)=-UYJ(IJ)/UXJ(IJ)
06350C FIND INTERSECTION WITH X-Z PLANE.
06355      UZJ2(IJ)=1.0
06360      UXJ2(IJ)=-UZJ(IJ)/UXJ(IJ)
06365      GO TO 94
06370 91 IF(UYJ(IJ).EQ.0.0) GO TO 92
06375C FIND INTERSECTION WITH Y-Z PLANE.
06380      UZJ1(IJ)=1.0
06385      UYJ1(IJ)=-UZJ(IJ)/UYJ(IJ)
06390C FIND INTERSECTION WITH X-Y PLANE.
06395      UXJ2(IJ)=1.0
06400      UYJ2(IJ)=-UXJ(IJ)/UYJ(IJ)
06405      GO TO 94
06410 92 IF(UZJ(IJ).EQ.0.0) GO TO 94
06415C FIND INTERSECTION WITH X-Z PLANE.
06420      UXJ1(IJ)=1.0
06425      UZJ1(IJ)=-UXJ(IJ)/UZJ(IJ)
06430C FIND INTERSECTION WITH Y-Z PLANE.
06435      UYJ2(IJ)=1.0
06440      UZJ2(IJ)=-UYJ(IJ)/UZJ(IJ)
06445 94 CONTINUE
06450      RR=SQRT(UXJ1(IJ)**2+UYJ1(IJ)**2+UZJ1(IJ)**2)
06455      UXJ1(IJ)=UXJ1(IJ)/RR
06460      UYJ1(IJ)=UYJ1(IJ)/RR
06465      UZJ1(IJ)=UZJ1(IJ)/RR
06470      RR=SQRT(UXJ2(IJ)**2+UYJ2(IJ)**2+UZJ2(IJ)**2)
06475      UXJ2(IJ)=UXJ2(IJ)/RR
06480      UYJ2(IJ)=UYJ2(IJ)/RR
06485      UZJ2(IJ)=UZJ2(IJ)/RR
06490 95 CONTINUE
06495      RETURN
06500 980 WRITE(6,981)
06505 981 FORMAT(/, ' **** ERROR IN WIRE INPUT')
06510      STOP
06515 990 WRITE(6,991)
06520 991 FORMAT(/, ' **** EPROR IN JUNCTION INPUT')
06525      STOP
06530      END
07940C -----
07950      SUBROUTINE ORDER(MCRD)
07960C DETERMINE THE ORDER IN WHICH THE Y MATRIX WAS GENERATED, AND
07970C CHECK FOR DIMENSIONAL CONSISTENCY BETWEEN THE Y MATRIX DATA FILE,
07980C AND THE INPUT DATA FILE.
07990      DIMENSION MORD(2)
08000      COMMON /BOT1/ NMODE,NPT,NBAND
08010      COMMON /BOT2/ NP,RL,YB(83),XB(83),YB1(82),XB1(82)

```

```

08020      COMMON /CAP1/ NC,XC,YC,ZC(2)
08025      COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
08030      COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
08032      1 XW1(100),YW1(100),ZW1(100)
08034      COMMON /WIRE3/ NW,INDW(6),RADW(5)
08040      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
08042      COMMON /EDG1/ NE,ZE(2),ZBE(10)
08050      READ(1) NMODE1,NP1,NC1,NPR1,NE1,NW1,NPW1,NJ1
08052      IF(NMODE.NE.NMODE1) GO TO 500
08054      IF(NP.NE.NP1) GO TO 500
08056      IF(NC.NE.NC1) GO TO 500
08058      IF(NPR.NE.NPR1) GO TO 500
08060      IF(NE.NE.NE1) GO TO 500
08062      IF(NW.NE.NW1) GO TO 500
08064      IF(NPW.NE.NPW1) GO TO 500
08066      IF(NJ.NE.NJ1) GO TO 500
08120C      MORD SPECIFIES THE ORDER IN WHICH CAPS AND/OR WIRES HAVE BEEN ADDED
08130C      TO THE BOT, INDEX 1 REFERS TO CAPS AND INDEX 2 REFERS TO WIRES.
08140C      MORD(I)=0 IF I HAS NOT BEEN ADDED
08150C      MORD(I)=N IF I HAS BEEN ADDED TO THE SYSTEM MATRIX, AND
08160C      IS LOCATED AT PSEUDO MODE NUMBER N.
08170      READ(1) MORD
08220      WRITE(6,4) MORD
08230 4      FORMAT(' THE Y MATRIX CONTAINS THE FOLLOWING ADDITIONS ( 0 IF ',
08240      1 ' NOT INCLUDED, OR CORRESPONDING MODE NUMBER IF PRESENT) ',//,
08250      2 ' 1X, 'CAPS', I4,/, '10X, 'WIRES', I3)
08260      RETURN
08270 500    WRITE(6,1)
08280 1      FORMAT(//, ' *** ERROR *** INPUT PARAMETERS DO NOT CHECK WITH ',
08290      1 ' THE PARAMETERS IN THE Y MATRIX FILE ',//,
08300      2 ' 6X, ' NMODE NP NC NPR NE NW NPW NJ ')
08310      WRITE(6,2) NMODE,NP,NC,NPR,NE,NW,NPW,NJ
08320 2      FORMAT(' INPUT ',8I6)
08330      WRITE(6,3) NMODE1,NP1,NC1,NPR1,NE1,NW1,NPW1,NJ1
08340 3      FORMAT(' Y FILE ',8I6)
08350      STOP
08360      END
08670C *****
08680      SUBROUTINE RBT(RBT,RBP,NT,TH,PHI)
08690C *****
08700C      COMPUTE BOT TRANSFER MATRICES FOR THETA AND PHI POLARIZATION.
08710      COMPLEX RBT(1),RBP(1),A6,U
08720      COMMON /WAVE/ WK
08730      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
08740      COMMON /BOT3/ DH(82),SV(82),CV(82)
08750      COMMON /BOT5/ T(160),TP(160),TZ(160)
08760      DIMENSION TH(1),PHI(1)
08770      DATA PI,U /3.14159265,(0.,1.) /
08780      DATA DTOR /0.017453292 /
08790      LS=NP-3
08800      NM=LS/2
08810      DO 156 K=1,NT

```

```

08820      K1=(K-1)*LS
08830      CS=COS(TH(K)*DTOR)
08840      SN=SIN(TH(K)*DTOR)
08850      CSPHI=COS(PHI(K)*DTOR)
08860      SNPHI=SIN(PHI(K)*DTOR)
08870      DO 300 J=1,NM
08880      J1=J+K1
08890      J2=J1+NM
08900      RBT(J1)=0.
08910      RBT(J2)=0.
08920      RBP(J1)=0.
08930      RBP(J2)=0.
08940C  CALCULATION OF TRANSFER MATRICES (EQS. 38-41).
08950C  NOTE, ONLY THE MODE INDEPENDENT PORTION IS COMPUTED HERE.
08960      DO 301 I=1,4
08970      I1=2*(J-1)+1
08980      I4=4*(J-1)+1
08990      A6=CEXP(U*BK*SN*(XB1(I1)*CSPHI+YB1(I1)*SNPHI))
09000C  THETA POLARIZED (EQS. 38-39).
09010      RBT(J1)=RBT(J1)+DH(I1)*T(I4)*(CV(I1)*CSPHI+SV(I1)*SNPHI)*A6
09020      RBT(J2)=RBT(J2)+DH(I1)*TZ(I4)*A6
09030C  PHI POLARIZED (EQS. 40-41).
09040      RBP(J1)=RBP(J1)+DH(I1)*T(I4)*(SV(I1)*CSPHI-CV(I1)*SNPHI)*A6
09050      301 CONTINUE
09060      RBT(J1)=CS*RBT(J1)
09070      RBT(J2)=-SN*RBT(J2)
09080      300 CONTINUE
09090      156 CONTINUE
09100      RETURN
09110      END
09111C  -----
09112      SUBROUTINE RCAP(RCT,RCP,NT,TH,PHI)
09113C
09114C  COMPUTE CAP TRANSFER MATRICES FOR THETA AND PHI POLARIZATION.
09115C
09116      COMPLEX RCT(1),RCP(1),A6,U
09117      COMMON /WAVE/ BK
09118      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
09119      COMMON /BOT3/ DH(82),SV(82),CV(82)
09120      COMMON /BOT5/ T(160),TP(160),TZ(160)
09121      COMMON /CAP1/ NC,XC,YC,ZC(2)
09122      COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
09123      COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
09124      COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
09125      COMMON /EDG1/ NE,ZE(2),ZBE(10)
09126      COMMON /EDG2/ TCE(10),TPCE(10),TBE(10),TPBE(10)
09127      DIMENSION TH(1),PHI(1)
09128      DATA PI,U /3.14159265,(0.,1.)/
09129      DATA DTOR /0.017453292/
09130      NM=(NP-3)/2
09131      LR=(NPR-3)/2
09132      LC=NC*NM*LR

```

```

09133      LE=NE+NM
09134      DO 156 K=1,NT
09135      K1=(K-1)*(2*LC+LE)
09136      CS=COS(TH(K)*DTOR)
09137      SN=SIN(TH(K)*DTOR)
09138      CSPHI=COS(PHI(K)*DTOR)
09139      SNPHI=SIN(PHI(K)*DTOR)
09140      J1=K1
09141      DO 300 JC=1,NC
09142      DO 300 JP=1,NM
09143      DO 300 JR=1,LR
09144      J1=J1+1
09145      J2=J1+LC
09146      RCT(J1)=0.
09147      RCT(J2)=0.
09148      RCP(J1)=0.
09149      RCP(J2)=0.
09150C  CALCULATION OF TRANSFER MATRICES.
09151      DO 301 IIP=1,4
09152      IP1=2*(JP-1)+IIP
09153      IP4=4*(JP-1)+IIP
09154      DO 301 IIR=1,4
09155      IR1=2*(JR-1)+IIR
09156      IR4=4*(JR-1)+IIR
09157      XX=XC+RHOC1(IR1)*RC1(IP1)*CPC(IP1)
09158      YY=YC+RHOC1(IR1)*RC1(IP1)*SPC(IP1)
09159      ZZ=ZC(JC)
09160      A6=CEXP(U*BK*(SN*(XX*CSPHI+YY*SNPHI)+ZZ*CS))
09161      AI=AC(IP1)*ABS(RHOC(IR1+1)**2-RHOC(IR1)**2)
09162C  THETA POLARIZED.
09163      RCT(J1)=RCT(J1)+AI*(CV(IP1)*CSPHI+SV(IP1)*SNPHI)*
09164      1 T(IP4)*TCT(IR4)*A6
09165      RCT(J2)=RCT(J2)+AI*(CPC(IP1)*CSPHI+SPC(IP1)*SNPHI)*
09166      1 TZ(IP4)*TCR(IR4)*A6
09167C  PHI POLARIZED.
09168      RCP(J1)=RCP(J1)+AI*(SV(IP1)*CSPHI-CV(IP1)*SNPHI)*
09169      1 T(IP4)*TCT(IR4)*A6
09170      RCP(J2)=RCP(J2)+AI*(SPC(IP1)*CSPHI-CPC(IP1)*SNPHI)*
09171      1 TZ(IP4)*TCR(IR4)*A6
09172 301 CONTINUE
09173      RCT(J1)=CS*RCT(J1)
09174      RCT(J2)=CS*RCT(J2)
09175 300 CONTINUE
09176      IF(NE.EQ.0) GO TO 156
09177C  EDGE TRANSFER MATRICES FOLLOW.
09178      J2=2*LC+K1
09179      DO 400 JC=1,NC
09180      DO 400 JP=1,NM
09181      J2=J2+1
09182      RCT(J2)=0.
09183      RCP(J2)=0.
09184      DO 401 IIP=1,4

```

```

09185      IP1=2*(JP-1)+IIP
09186      IP4=4*(JP-1)+IIP
09187      DO 401 IIRZ=1,2
09188      IR1=2*LR+IIRZ
09189      IR4=2*(JC-1)+IIRZ
09190      IZ1=2*(JC-1)+IIRZ
09191      IZ4=2*(JC-1)+IIRZ
09192      XX=XC+RHOC1(IR1)*RC1(IP1)*CPC(IP1)
09193      YY=YC+RHOC1(IR1)*RC1(IP1)*SPC(IP1)
09194      ZZ=ZC(JC)
09195      A6=CEXP(U*BK*(SN*(XX*CSPHI+YY*SNPHI)+ZZ*CS))
09196      AI=AC(IP1)*ABS(RHOC(IR1+1)**2-RHOC(IR1)**2)
09197C  EDGE(CAP)
09198C  THETA POLARIZED.
09199      RCT(J2)=RCT(J2)+CS*AI*(CPC(IP1)*CSPHI+SPC(IP1)*SNPHI)*
09200      1 TZ(IP4)*TCE(IR4)*A6
09201C  PHI POLARIZED.
09202      RCP(J2)=RCP(J2)+AI*(SPC(IP1)*CSPHI-CPC(IP1)*SNPHI)*
09203      1 TZ(IP4)*TCE(IR4)*A6
09204      XX=XB1(IP1)
09205      YY=YB1(IP1)
09206      ZZ=ZB1(IZ1)
09207      A6=CEXP(U*BK*(SN*(XX*CSPHI+YY*SNPHI)+ZZ*CS))
09208      AI=DH(IP1)*ABS(ZE(JC)-ZC(JC))/2.
09209C  EDGE(BOT).
09210C  THETA POLARIZED.
09211      RCT(J2)=RCT(J2)-SN*AI*TZ(IP4)*TBE(IZ4)*A6
09212  401 CONTINUE
09213  400 CONTINUE
09214  156 CONTINUE
09215      RETURN
09216      END
09222C -----
09223      SUBROUTINE RWIRE(RWT,RWP,NT,TH,PHI)
09224C
09225C  COMPUTE WIRE-JUNCTION TRANSFER MATRICES.
09230C
09240      COMPLEX RWT(1),RWP(1),A6,U
09245      COMMON /WAVE/ BK
09250      COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
09260      1 XW1(100),YW1(100),ZW1(100)
09270      COMMON /WIRE2/ DHW(100),DXW(100),DYW(100),DZW(100)
09280      COMMON /WIRE3/ NW,INDW(6),RADW(5)
09290      COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
09300      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
09310      COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
09320      COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
09330      COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
09340      COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
09350      DIMENSION TH(1),PHI(1)
09360      DATA PI,U /3.14159265,(0.,1.)/
09360      DATA DTOR /0.017453292/

```

```

09380      LWJ=LW+NJ
09390      DO 156 K=1,NT
09400      K1=(K-1)*LWJ
09410      CS=COS(TH(K)*DTOR)
09420      SN=SIN(TH(K)*DTOR)
09430      CSPHI=COS(PHI(K)*DTOR)
09440      SNPHI=SIN(PHI(K)*DTOR)
09450      DO 300 J=1,LW
09455      J1=J+K1
09460      RWT(J1)=0.
09465      RWP(J1)=0.
09470      DO 301 I=1,4
09475      I1=INDTW(J)+I-1
09480      I4=4*(J-1)+I
09485      A6=CEXP(U*BK*(SN*(XW1(I1)*CSPHI+YW1(I1)*SNPHI)+ZW1(I1)*CS))
09490      RWT(J1)=RWT(J1)+(CS*(DXW(I1)*CSPHI+DYW(I1)*SNPHI)-
09495      1 DZW(I1)*SN)*TJ(I4)*A6
09500      RWP(J1)=RWP(J1)+(-DXW(I1)*SNPHI+DYW(I1)*CSPHI)*TW(I4)*A6
09505      301 CONTINUE
09510      300 CONTINUE
09515      IF(NJ.EQ.0) GO TO 156
09520      J1=LW+K1
09525      DO 400 J=1,NJ
09530      J1=J1+1
09535      RWT(J1)=0.
09540      RWP(J1)=0.
09545      DO 401 I=1,2
09550      I1=INDTJ(J)+I-1
09555      I2=2*(J-1)+I
09560      A6=CEXP(U*BK*(SN*(XW1(I1)*CSPHI+YW1(I1)*SNPHI)+ZW1(I1)*CS))
09565      RWT(J1)=RWT(J1)+(CS*(DXW(I1)*CSPHI+DYW(I1)*SNPHI)-
09570      1 DZW(I1)*SN)*TJ(I2)*A6
09575      RWP(J1)=RWP(J1)+(-DXW(I1)*SNPHI+DYW(I1)*CSPHI)*TJ(I2)*A6
09580      401 CONTINUE
09585      DO 402 JJ=1,4
09590      ALPHA=PI*((JJ-1)/2.+0.25)
09595      SA=SIN(ALPHA)
09600      CA=COS(ALPHA)
09605      RA=(RADJ(J)+RADD(J))/2.
09610      XA=XJ(J)+RA*(CA*UXJ1(J)+SA*UXJ2(J))
09615      YA=YJ(J)+RA*(CA*UYJ1(J)+SA*UYJ2(J))
09620      ZA=ZJ(J)+RA*(CA*UZJ1(J)+SA*UZJ2(J))
09625      DT=CA*(UXJ1(J)*CS+CSPHI+UYJ1(J)*CS*SNPHI-UZJ1(J)*SN)+
09630      1 SA*(UXJ2(J)*CS+CSPHI+UYJ2(J)*CS*SNPHI-UZJ2(J)*SN)
09635      DP=CA*(-UXJ1(J)*SNPHI+UYJ1(J)*CSPHI)+
09640      1 SA*(-UXJ2(J)*SNPHI+UYJ2(J)*CSPHI)
09645      RR=(RADJ(J)-RADD(J))/2.
09650      A6=CEXP(U*BK*(SN*(XA*CSPHI+YA*SNPHI)+ZA*CS))
09655      RWT(J1)=RWT(J1)+RR/4.*DT*A6
09660      RWP(J1)=RWP(J1)+RR/4.*DP*A6
09665      402 CONTINUE
09670      400 CONTINUE

```

09670 156 CONTINUE
09680 RETURN
09690 END
09700/EOR

BOONSS

PRECEDING PAGE BLANK

```

00100BOT5,P30,T100,CH14C000,STCX3.
00110ACCOUNT,M03255,BOT2A.
00120BANNERS(OUTPUT)*J. PUTNAM*DEPT 220*BLD 110-4**
00130ATTACH(INFIL=YSSX)
00140FTN(R=0,OPT=1)
00164LDSET(PRESET=INDEF)
00170LG0.
00190/EOR
00200      PROGRAM BOTSCB(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
00210      1 INFIL,TAPE1=INFIL)
00220C
00230C BOTSCB IS THE BOT BI-STATIC SCATTERING CODE.
00240C
00250C UNIT 5 IS THE CARD READER.
00260C UNIT 6 IS THE LINE PRINTER.
00270C UNIT 1 IS A DISK FILE CONTAINING THE Y MATRIX.
00280C
00285      COMPLEX U
00290      COMPLEX STT(91),SPP(91),STP(91),SPT(91)
00300      COMPLEX CBT(1680),CBP(1680),CMT(59),CMP(59),CCT(96),CCP(96)
00310      COMPLEX ESC(3),ESCP(3),HSCT(3),HSCP(3)
00320      COMPLEX Y(2304)
00325      COMPLEX RBT(82),RBP(82),RWT(59),RWP(59),RCT(96),RCP(96)
00330      COMMON /WAVE/ BK
00340      COMMON /BOT1/ NMODE,NPT,NBAND
00350      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
00360      COMMON /BOT3/ D4(82),SV(82),CV(82)
00370      COMMON /BOT5/ T(160),TP(160),TZ(160)
00380      COMMON /BOT6/ IEDGE,IUNIF
00382      COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
00384      1 XW1(100),YW1(100),ZW1(100)
00385      COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
00386      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RAOD(10)
00387      COMMON /CAP1/ NC,XC,YC,ZC(2)
00388      COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
00389      COMMON /EDG1/ NE,ZE(21),ZBE(10)
00390      COMMON /INT/ N,RH02,ZP
00400      DIMENSION THS(73),PHIS(73)
00410      DIMENSION ANG(6),IPLANE(6),ANG1(6),ANG2(6)
00420      DIMENSION MORD(2)
00430      U=(0.,1.)
00440      ETA=376.707
00450      PI=3.14159265
00460      DTOR=PI/180.
00462      READ(5,1) BK
00463      1 FORMAT(E15.7)
00464      WRITE(6,2) BK
00465      2 FORMAT(9H1      BK,/,E15.7)
00470C
00480      CALL BOTIN
00490C
00500      CALL CAPIN

```

```

00510C      CALL WIREIN
00520C
00522C
00530C READ INPUTS, DESCRIBING THE INCIDENT FIELD AND SCATTERING PLANES.
00540      READ(5,44) NANG,NT,PHI1,THI
00550 44      FORMAT(2I3,2F8,4)
00560      READ(5,53) (ANG(I),I=1,NANG)
00570 53      FORMAT(10F8,4)
00580C IPLANE = 1 FOR PHI FIXED,
00590C           2 FOR THETA FIXED,
00600      READ(5,54) (IPLANE(I),I=1,NANG)
00610 54      FORMAT(10I8)
00620      READ(5,53) (ANG1(I),I=1,NANG)
00630      READ(5,53) (ANG2(I),I=1,NANG)
00640      WRITE(6,3) NANG,NT
00650 3      FORMAT(//,25H NUMBER OF FIXED ANGLES =,I3,//
00660      1 35H NUMBER OF ANGLES PER FIXED ANGLE =,I4)
00670      WRITE(6,4) (ANG(I),IPLANE(I),ANG1(I),ANG2(I),I=1,NANG)
00680 4      FORMAT(//,45H FIXED ANGLE CODE VARIABLE ANGLE RANGE,
00690      1 //,(F8.1,9X,I2,6X,F8.1,34 - ,F8.1))
00700      LS=NP-3
00710      NM=LS/2
00720      NM1=NM+1
00730      NM4=NM+4
00740      LSS=LS+LS
00750      LR=(NPR-3)/2
00760      LC=NC+NM+LR
00770      LC2=LC+2
00780      LE=NE+NM
00790      LSC=LS*(LC2+LE)
00800      LWJ=LW+NJ
00810      LSM=LS*LWJ
00820      LCC=(LC2+LE)**2
00830      LCM=(LC2+LE)*LWJ
00840      LWM=LWJ**2
00850      BKL=BK+BL
00860      P1=(BK*BK*ETA/2./PI)**2/4./PI
00870      KMODE=2*NMODE-1
00880      CALL ZERO(LS*KMODE,CBT)
00890      CALL ZERO(LS*KMODE,CBP)
00900      IF(NC.NE.0) CALL ZERO(LC2+LE,CCT)
00910      IF(NC.NE.0) CALL ZERO(LC2+LE,CCP)
00920      IF(NPM.NE.0) CALL ZERO(LWJ,CWT)
00930      IF(NPM.NE.0) CALL ZERO(LWJ,CWP)
00940      IF(NPM.NE.0) CALL ZERO(LWJ,CW)
00950      CALL ORDER(MORD)
00960C CALCULATE CURRENTS, EQ. 10
00970      CALL ROOT(RBT,RBP,1,THI,PHI1)
00980      IF(NC.NE.0) CALL RCAP(RCT,RCP,1,THI,PHI1)
00990      IF(NPM.NE.0) CALL RWIRE(RWT,RWP,1,THI,PHI1)
01000      NUMB=0
01010 50      CONTINUE

```

```

00920      READ(1) M,N
00930      IF(IECF(1)) 37,60
00940 60     NUMB=NUMB+1
00942      IC=(N+NM0DE-1)*LS+1
00950C DETERMINE SUBMATRIX TYPE, AND CALCULATE CURRENTS.
00960      IF(N.LT.NM0DE) GO TO 61
00970      IF(N.EQ.MORD(1)) GO TO 62
00980      IF(N.EQ.MORD(2)) GO TO 63
00990      GO TO 50
00992 61     DNT=2.*BL*SINC(-M*PI+BKL*COS(THI+DTOR))
00994      DNZ=DNT-2.*BL*(-1)**N*SINC(BKL*COS(THI+DTOR))
01010      IF(N.LT.NM0DE) GO TO 71
01020      IF(N.EQ.MORD(1)) GO TO 72
01030      IF(N.EQ.MORD(2)) GO TO 73
01040      GO TO 50
01050 62     IF(N.LT.NM0DE) GO TO 74
01060      IF(N.EQ.MORD(1)) GO TO 75
01070      IF(N.EQ.MORD(2)) GO TO 76
01080      GO TO 50
01090 63     IF(N.LT.NM0DE) GO TO 77
01100      IF(N.EQ.MORD(1)) GO TO 78
01110      IF(N.EQ.MORD(2)) GO TO 79
01120      GO TO 50
01130C BOT-BOT.
01140 71     READ(1) (Y(I),I=1,LSS)
01170      CALL MULTYR(LS,LS,Y,DNT,DNZ,RBT,CBT(IC))
01175      CALL MULTYR(LS,LS,Y,DNT,DNZ,RBP,CBP(IC))
01180      GO TO 50
01190C CAP-BOT.
01200 72     READ(1) (Y(I),I=1,LSC)
01210      CALL MULTYR(LC2+LE,LS,Y,DNT,DNZ,RBT,CCT)
01215      CALL MULTYR(LC2+LE,LS,Y,DNT,DNZ,RBP,CCP)
01220      GO TO 50
01230C WIRE-BOT.
01240 73     READ(1) (Y(I),I=1,LSW)
01250      CALL MULTYR(LWJ,LS,Y,DNT,DNZ,RBT,CWT)
01255      CALL MULTYR(LWJ,LS,Y,DNT,DNZ,RBP,CWP)
01260      GO TO 50
01270C BOT-CAP.
01280 74     READ(1) (Y(I),I=1,LSC)
01290      CALL MULTYR(LS,LC2+LE,Y,1.0,1.0,RCT,CBT(IC))
01295      CALL MULTYR(LS,LC2+LE,Y,1.0,1.0,RCP,CBP(IC))
01300      GO TO 50
01310C CAP-CAP.
01320 75     READ(1) (Y(I),I=1,LCC)
01330      CALL MULTYR(LC2+LE,LC2+LE,Y,1.0,1.0,RCT,CCT)
01335      CALL MULTYR(LC2+LE,LC2+LE,Y,1.0,1.0,RCP,CCP)
01340      GO TO 50
01350C WIRE-CAP.
01360 76     READ(1) (Y(I),I=1,LWJ)
01370      CALL MULTYR(LWJ,LC2+LE,Y,1.0,1.0,RCT,CWT)
01375      CALL MULTYR(LWJ,LC2+LE,Y,1.0,1.0,RCP,CWP)

```

```

01380      GO TO 50
01390C BOT-WIRE
01400 77 READ(1) (Y(I),I=1,LSW)
01410      CALL MULTYR(LS,LWJ,Y,1.0,1.0,RWT,CBT(IC))
01415      CALL MULTYR(LS,LWJ,Y,1.0,1.0,RWP,CBP(IC))
01420      GO TO 50
01430C CAP-WIRE
01440 78 READ(1) (Y(I),I=1,LCW)
01450      CALL MULTYR(LC2+LE,LWJ,Y,1.0,1.0,RWT,CCT)
01455      CALL MULTYR(LC2+LE,LWJ,Y,1.0,1.0,RWP,CCP)
01460      GO TO 50
01470C WIRE-WIRE
01480 79 READ(1) (Y(I),I=1,LWJ)
01490      CALL MULTYR(LWJ,LW+NJ,Y,1.0,1.0,RWT,CWT)
01495      CALL MULTYR(LWJ,LW+NJ,Y,1.0,1.0,RWP,CWP)
01500      GO TO 50
01550C ALL Y SUBMATRICES HAVE BEEN READ.
01560 37 CONTINUE
01562      WRITE(6,114) NUMB
01564 114 FORMAT(//,IX,I4,17H SUBMATRICES READ)
01630C *** END CALCULATION OF CURRENTS ***
01630      DO 90 IANG=1,NANG
01660C INITIALIZE ARRAYS, AND SET THE SCATTERING ANGLES.
01670      IPLAN=IPLANE(IANG)
01675      DT=(ANG2(IANG)-ANG1(IANG))/(NT-1)
01680      DO 100 K=1,NT
01685      STT(K)=0.0
01690      SPP(K)=0.0
01695      STP(K)=0.0
01700      SPT(K)=0.0
01710      PHIS(K)=ANG(IANG)*(2-IPLAN)+(ANG1(IANG)+DT*(K-1))*(IPLAN-1)
01720 100 THS(K)=ANG(IANG)+(IPLAN-1)+(ANG1(IANG)+DT*(K-1))*(2-IPLAN)
01725      WRITE(6,403) PHI,THI
01730 403 FORMAT(1H1,22X,34HBI-STATIC RCS(DB). INCIDENT PHI =,F6.1,
01735      1 4H THETA =,F6.1,/,20X,45(1H-),/,
01740      2 5H PHI THETA 00 00 00
01745      3 5H 00,/, THETA -- // -/
01750      4 5H+ /-,/)
01755      5H
01770C CALCULATE RCS.
01780      DO 110 K=1,NT
01790      CALL RBDT(RBT,RBP,1,THS(K),PHIS(K))
01800      IC=0
01810      DO 115 NM=1,KMODE
01820      N=-NM*NM+NM
01830C CALCULATE DN AS A FUNCTION OF THETA (EQ. 41).
01840      DMT=2.*BL*SINC(N*PI+BKL*COS(THS(K)+DTOR))
01850      DMZ=DMT-2.*BL*(-1)**N*SINC(BKL*COS(THS(K)+DTOR))
01860      DO 105 J=1,LS
01870      IC=IC+1
01880      DN=DMT
01890      IF(J.GT.NM) DN=DMZ

```

```

01895 STT(K)=STT(K)+04*RBT(J)*CBT(IC)
01900 SPP(K)=SPP(K)+04*RBP(J)*CBP(IC)
01905 STP(K)=STP(K)+04*RBT(J)*CBP(IC)
01910 SPT(K)=SPT(K)+04*RBP(J)*CBT(IC)
01920 105 CONTINUE
01921 115 IF(LC.EQ.0) GO TO 106
01922 CALL RCAP(RCT,RCP,1,THS(K),PHIS(K))
01924 CALL MULTRC(LC2+LE,RCT,CCT,STT(K))
01925 CALL MULTRC(LC2+LE,RCP,CCP,SPP(K))
01926 CALL MULTRC(LC2+LE,RCT,CCP,STP(K))
01927 CALL MULTRC(LC2+LE,RCP,CCT,SPT(K))
01928 106 CONTINUE
01930 IF(NPW.EQ.0) GO TO 107
01932 CALL RWIRE(RWT,RWP,1,THS(K),PHIS(K))
01934 CALL MULTRC(LWJ,RWT,CWT,STT(K))
01935 CALL MULTRC(LWJ,RWP,CWP,SPP(K))
01936 CALL MULTRC(LWJ,RWT,CWP,STP(K))
01937 CALL MULTRC(LWJ,RWP,CWT,SPT(K))
01938 107 CONTINUE
01939 X2=CABS(STT(K))
01940 X2=DBCON(P1*X2*X2)
01941 X3=CABS(SPP(K))
01942 X3=DBCON(P1*X3*X3)
01943 X4=CABS(STP(K))
01944 X4=DBCON(P1*X4*X4)
01945 X5=CABS(SPT(K))
01946 X5=DBCON(P1*X5*X5)
01947 WRITE(6,113) PHIS(K),THS(K),X2,X3,X4,X5
01948 113 FORMAT(1X,2F8.1,4F12.2)
01949 110 CONTINUE
02000 90 CONTINUE
02010C *** END CALCULATION OF RCS ***
02020C LIST AND PLOT CURRENTS.
02030C WRITE(6,301)
02031 301 FORMAT(32H1 THETA POLARIZED INCIDENT FIELD,/)
02060 CALL LPCUR(NMODE,NM,CBT,LW,NJ,CWT,LC,LE,CCT)
02065 WRITE(6,302)
02070 302 FORMAT(30H1 PHI POLARIZED INCIDENT FIELD,/)
02075 CALL LPCUR(NMODE,NM,CBP,LW,NJ,CWP,LC,LE,CCP)
02080C *** END LISTING AND PLOTTING OF CURRENTS ***
02210C
02220C NEAR FIELD ANALYSIS FOLLOWS.
02230C
02235C NOTE, ESC IS FOR ELECTRIC FIELDS
02240C HSC IS FOR MAGNETIC FIELDS
02245C INDICES 1,2,AND 3 ARE FOR THE X,Y,AND Z COMPONENTS OF THE
02250C FIELDS, RESPECTIVELY.
02255C THE T ENDING IS FOR THETA INCIDENT POLARIZATION
02260C THE P ENDING IS FOR PHI INCIDENT POLARIZATION.
02265C
02270 400 READ(5,44) NTEST
02271 IF(NTEST.EQ.0) GO TO 900

```

```

02275      WRITE(6,404)
02280 404      FORMAT(1H1,/,21H NEAR FIELD ANALYSIS,/)
02285      1 59X,7HE-FIELD,37X,7HH-FIELD,/,
02290      2 43X,39(1H-),5X,39(1H-),/,
02295      3 35X,16HFIELD COMPONENTS,28X,16HFIELD COMPONENTS,/,
02300      4 24H ZTEST YTEST XTEST,
02305      5 11X,2(14X,6H X,7X,5H Y,9X,3H Z ),/)
02310      DO 400 ITEST=1,NTEST
02315      READ(5,53) ZTEST,YTEST,XTEST
02320      HSCCT(1)=0.
02325      HSCCT(2)=0.
02330      HSCCT(3)=0.
02335      HSCCT(1)=0.
02340      HSCCT(2)=0.
02345      HSCCT(3)=0.
02350      HSCCP(1)=0.
02355      HSCCP(2)=0.
02360      HSCCP(3)=0.
02365      HSCCP(1)=0.
02370      HSCCP(2)=0.
02375      HSCCP(3)=0.
02380      HSCCP(1)=0.
02385      HSCCP(2)=0.
02390      HSCCP(3)=0.
02395      HSCCP(1)=0.
02400      HSCCP(2)=0.
02405      HSCCP(3)=0.
02410      BOT CONTRIBUTION.
02415      STEP THROUGH MODES.
02420      N=NMODE
02425      DO 430 NM=1,KMODE
02430      N=N+1
02435      CALCULATE N-TH MODAL CURRENT COEFFICIENTS (EQS. 67 & 71).
02440      CALL NEAR8(XTEST,YTEST,ZTEST,Y)
02445      CALCULATE AND SUM Y*CUR, USING THE FIRST 6 ROWS OF Y, YIELDING
02450      THE 3 FIELD COMPONENTS FOR BOTH ELECTRIC AND MAGNETIC NEAR FIELDS,
02455      FOR BOTH INCIDENT POLARIZATIONS, (EQS. 67 & 71).
02460      DO 420 K=1,3
02465      I1=(N+NMODE-1)+LS
02470      J1=K-6
02475      DO 420 I=1,LS
02480      I1=I1+1
02485      J1=J1+6
02490      J2=J1+3
02495      ESCT(K)=ESCT(K)+Y(J1)*CBT(I1)
02500      HSCT(K)=HSCT(K)+Y(J2)*CBT(I1)
02505      ESCP(K)=ESCP(K)+Y(J1)*CBP(I1)
02510      HSCP(K)=HSCP(K)+Y(J2)*CBP(I1)
02515      420 CONTINUE
02520      430 CONTINUE
02525      IF(NC.EQ.0) GO TO 450
02530      CAP CONTRIBUTION.
02535      CALL NEAR8(XTEST,YTEST,ZTEST,Y)
02540      DO 440 K=1,3
02545      J1=K-6
02550      DO 440 I=1,LC2
02555      J1=J1+6
02560      J2=J1+3

```

```

C2508      ESCT(K)=ESCT(K)+Y(J1)*CCT(I)
C2509      HSCT(K)=HSCT(K)+Y(J2)*CCT(I)
C2510      ESCP(K)=ESCP(K)+Y(J1)*CCP(I)
C2511      HSCP(K)=HSCP(K)+Y(J2)*CCP(I)
C2512      440 CONTINUE
C2513      450 IF(INPW.EQ.0) GO TO 470
C2514      WIRE CONTRIBUTION.
C2515      CALL NEARM(XTEST,YTEST,ZTEST,Y)
C2516      DO 460 K=1,3
C2517      J1=K-6
C2518      DO 460 I=1,LW
C2519      J1=J1+6
C2520      J2=J1+3
C2521      ESCT(K)=ESCT(K)+Y(J1)*CWT(I)
C2522      HSCT(K)=HSCT(K)+Y(J2)*CWT(I)
C2523      ESCP(K)=ESCP(K)+Y(J1)*CWP(I)
C2524      HSCP(K)=HSCP(K)+Y(J2)*CWP(I)
C2525      460 CONTINUE
C2526      470 WRITE(6,505) ZTEST,YTEST,XTEST
C2527      505 FORMAT(/,1X,3F8.4)
C2528      WRITE(6,506) (CABS(ESCT(I)),I=1,3),(CABS(HSCT(I)),I=1,3)
C2529      506 FORMAT(26X,17H) INC POL E-FIELD,3E13.4,5X,3E13.4,/,1H+,25X,1H-)
C2530      WRITE(6,507) (CABS(ESCP(I)),I=1,3),(CABS(HSCP(I)),I=1,3)
C2531      507 FORMAT(26X,17H) INC POL E-FIELD,3E13.4,5X,3E13.4,/,1H+,25X,1H/)
C2532      490 CONTINUE
C2533      900 STOP
C2534      END
-----
C2821C      SUBROUTINE ZERO(N,A)
C2822C      A=0.
C2823C      COMPLEX A(1)
C2824C      DO 100 I=1,N
C2825C      A(I)=0.0
C2826C      100 CONTINUE
C2827C      RETURN
C2828C      END
-----
C2830C      SUBROUTINE MULTYR(NI,NJ,Y,DT,DZ,V,C)
C2831C      UPDATE CURRENT ARRAY C BY Y * (DT,DZ) * V.
C2832C      COMPLEX S,Y(1),V(1),C(1)
C2833C      NJ2=NJ/2
C2834C      DO 100 I=1,NI
C2835C      S=0.0
C2836C      IND=1
C2837C      DO 50 J=1,NJ
C2838C      D=DT
C2839C      IF(J.GT.NJ2) D=DZ
C2840C      50 S=S+Y(IND)*D*V(J)
C2841C      IND=IND+NI
C2842C      C(I)=C(I)+S

```

```

02960 100 CONTINUE
02970 RETURN
02980 END
03100C -----
03110 SUBROUTINE MULTRC(NI,R,C,G)
03120C
03130C UPDATE G BY R * C.
03140C
03150C   COMPLEX G,R(1),C(1)
03160C   DO 100 I=1,NI
03170C     G=G+R(I)*C(I)
03180 100 CONTINUE
03190 RETURN
03200 END
03210C *****
03220C FUNCTION SINC(X)
03230C *****
03240C   SINC=1.0
03250C   IF(X.NE.0.0) SINC=SIN(X)/X
03260C   RETURN
03270C   END
03280C *****
03290C FUNCTION DBCON(X)
03300C *****
03310C RETURN X IN DB.
03320C   DBCON=-1000.0
03330C   IF(X.EQ.0.0) RETURN
03340C   DBCON=10.*ALOG10(X)
03350C   RETURN
03360C   END
03370C -----
03380C SUBROUTINE BOTIN
03390C
03400C READ BOT COORDINATES AND COMPUTE BOT SEGMENT ARRAYS.
03410C
03420C   COMMON /BOT1/ NMODE,NPT,NBAND
03430C   COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
03440C   COMMON /BOT3/ DH(82),SV(82),CV(82)
03450C   COMMON /BOT5/ T(160),TP(160),TZ(160)
03460C   COMMON /BOT6/ IEDGE,IUNIF
03470C   READ(5,49) NMODE,NPT,NBAND
03480C   READ(5,49) NP
03490C 49  FORMAT(3I3)
03500C   WRITE(6,48) NMODE,NPT,NBAND,NP
03510C 48  FORMAT(32H      NMODE      NPT      NBAND      NP,/,4I8)
03520C   READ(5,53) (YB(I),I=1,NP)
03530C   READ(5,53) (XB(I),I=1,NP)
03540C 53  FORMAT(10F8.4)
03550C   WRITE(6,55)
03560C 55  FORMAT(7,3H YB)
03570C   WRITE(6,46) (YB(I),I=1,NP)
03580C 46  FORMAT(1X,10F8.4)

```

```

03480 WRITE(6,56)
03485 56 FORMAT(//,3H XB)
03490 WRITE(6,46)(XB(I),I=1,NP)
03495C PLOT THE BODY COORDINATES.
03500 CALL PLOT8(XB,YB,NP,41)
03505 READ(5,53) BL
03510 WRITE(6,47) BL
03515 47 FORMAT(//,21H HALF-LENGTH OF BOT =,F12.4)
03520C CHECK IF BOT CONTAINS AN ODD NUMBER OF POINTS.
03525 IF(MOD(NP,2).NE.1) GO TO 980
03530C CHECK IF GENERATING CURVE IS OPEN OR CLOSED.
03535 IEDE=1
03540 IF((YB(1)-YB(NP)).NE.0..OR.(XB(1)-XB(NP)).NE.0.) GO TO 10
03545 IEDE=0
03550 YB(NP+1)=YB(2)
03555 XB(NP+1)=XB(2)
03560 YB(NP+2)=YB(3)
03565 XB(NP+2)=XB(3)
03570 NP=NP+2
03575 WRITE(6,66) NP
03580 66 FORMAT(//,39H BOT GENERATING CURVE IS CLOSED. NP = ,I3)
03585 1C CONTINUE
03590C COMPUTATION OF BODY SEGMENT PARAMETERS.
03595 DO 57 I=2,NP
03600 I2=I-1
03605 RR1=YB(I)-YB(I2)
03610 RR2=XB(I)-XB(I2)
03615 DH(I2)=SQRT(RR1*RR1+RR2*RR2)
03620 XB(I2)=.5*(XB(I)+XB(I2))
03625 YB(I2)=.5*(YB(I)+YB(I2))
03630 SV(I2)=RR1/DH(I2)
03635 CV(I2)=RR2/DH(I2)
03640 57 CONTINUE
03645C CHECK IF BOT SEGMENTATION IS UNIFORM.
03650 IUNIF=0
03655 NP1=NP-1
03660 DO 60 I=2,NP1
03665 RR1=DH(I)/DH(1)
03670 IF(RR1.LT.0.99 .OR. RR1.GT.1.01) GO TO 20
03675 60 CONTINUE
03680 IUNIF=1
03685 WRITE(6,67)
03690 67 FORMAT(//,1 BOT GENERATING CURVE HAS UNIFORM SEGMENTATION')
03695 20 CONTINUE
03700C COMPUTATION OF TRIANGLE FUNCTIONS T.
03705 NM=(NP-3)/2
03710 DO 74 J=1,NM
03715 J2=2*(J-1)+1
03720 J3=J2+1
03725 J4=J3+1
03730 J5=J4+1
03735 J6=4*(J-1)+1

```

```

03740 J7=J6+1
03745 J8=J7+1
03750 J9=J8+1
03755 DEL1=DM(J2)+DM(J3)
03760 DEL2=DM(J4)+DM(J5)
03765 TP(J6)=1./DEL1
03770 TP(J7)=1./DEL1
03775 TP(J8)=-1./DEL2
03780 TP(J9)=-1./DEL2
03785 T(J6)=DM(J2)/2./DEL1
03790 T(J7)=(DM(J2)+DM(J3)/2.)/DEL1
03795 T(J8)=(DM(J4)/2.+DM(J5))/DEL2
03800 T(J9)=DM(J5)/2./DEL2
03805 74 CONTINUE
03810 NM4=NM+4
03815 DO 75 J=1,NM4
03820 75 TZ(J)=T(J)
03825 IF(EDGE.EQ.0) GO TO 76
03830 TZ(1)=2.-T(1)
03835 TZ(2)=2.-T(2)
03840 TZ(NM4-1)=2.-T(NM4-1)
03845 TZ(NM4)=2.-T(NM4)
03850 76 CONTINUE
03855 RETURN
03860 980 WRITE(6,981)
03865 981 FORMAT(/, '**** ERROR IN BOT INPUT')
03870 STOP
03875 END
04410C *****
04420 SUBROUTINE PLOTB(X,Y,N,NR)
04430C *****
04440C WRITTEN 2/14/74 BY J. M. PUTNAM DEPT 220 X23877
04450C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
04460C N IS THE NUMBER OF POINTS TO BE PLOTTED.
04470C NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
04480C NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
04490C NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
04500C
04510C REAL X(1),Y(1),HEAD(10)
04520C INTEGER LINE(101),BLANK,STAR,PLUS
04530C DATA BLANK,STAR,PLUS /1H*,1H+,1H+/
04540C NC=51
04550C N10=(NC-1)/10
04560C WRITE(6,500)
04570C 500 FORMAT(/,44H BODY COORDINATES, + INDICATES TRIANGLE PEAK)
04580C WRITE(6,504)
04590C XMIN=X(1)
04600C XMAX=X(1)
04610C YMIN=Y(1)
04620C YMAX=Y(1)
04630C
04640C

```

```

C4650      DO 6 I=1,N
C4660      IF(X(I).LT.XMIN) XMIN=X(I)
C4670      IF(X(I).GT.XMAX) XMAX=X(I)
C4680      IF(Y(I).LT.YMIN) YMIN=Y(I)
C4690      IF(Y(I).GT.YMAX) YMAX=Y(I)
C4700  6    CONTINUE
C4710      DEL=XMAX-XMIN
C4720      IF(YMAX-YMIN.GT.DEL) DEL=YMAX-YMIN
C4730      XMAX=XMIN+DEL
C4740      YMAX=YMIN+DEL
C4750      DO 5 I=1,N10
C4760      Z=I
C4770  5    HEAD(I)=(XMAX-XMIN)*Z/N10+XMIN
C4780      DY=(YMAX-YMIN)/(NR-1)
C4790      IEDGE=1
C4800      IF(X(1).EQ.X(N) .AND. Y(1).EQ.Y(N)) IEDGE=0
C4810      Z=YMAX+DY
C4820      YL=Z-DY/2.
C4830      DO 7 J=1,NR
C4840      DO 8 K=1,NC
C4850  8    LINE(K)=BLANK
C4860      Z=Z-DY
C4870      YU=YL
C4880      YL=Z-DY/2.
C4890      DO 9 I=1,N
C4900      IF(Y(I).GE.YU) GO TO 9
C4910      IF(Y(I).LT.YL) GO TO 9
C4920      K=(X(I)-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
C4930      IF(K.GT.NC) K=NC
C4940      LINE(K)=STAR
C4950      IF(MOD(I,2).EQ.1) LINE(K)=PLUS
C4960      IF(IEEDGE.EQ.C) GO TO 9
C4970      IF(I.EQ.1 .OR. I.EQ.N) LINE(K)=STAR
C4980  9    CONTINUE
C4990      WRITE(6,508) Z,(LINE(K),K=1,NC)
C5000  7    CONTINUE
C5010      WRITE(6,504)
C5020      WRITE(6,5002)
C5030      WRITE(6,5007) XMIN,(HEAD(I),I=1,N10)
C5040      RETURN
C5050  504  FORMAT (1X,14(1H-),1H.,10(5H----.),1H-)
C5060  507  FORMAT(10X,11(F10.4))
C5070  508  FORMAT(1X,F12.4,1X,1HI,51A1,1HI)
C5080  3002 FORMAT(4X,7HVH / XH,4X,1HI,5(9X,1HI))
C5090      END
C5100C -----
C5102C      SUBROUTINE CAPIN
C5104C
C5106C      READ CAP INPUTS AND COMPUTE CAP ARRAYS.
C5108C
C5110      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
C5112      COMMON /CAP1/ NC,XC,YC,ZC(2)

```

```

05114 COMMON /CAP2/ NPR, RHOC(21), RHOC1(20), DRHOC(20)
05116 COMMON /CAP3/ TCR(36), TCT(36), TPCR(36), TPC1(36)
05118 COMMON /CAP4/ RC(83), RC1(82), AC(82), CPC(82), SPC(82)
05120 COMMON /EDG1/ NE, ZE(2), ZBE(10)
05122 COMMON /EDG2/ TCE(10), TPCE(10), T9E(10), TPBE(10)
05124 READ(5,1) NC, NPR, NE
05126 1 FORMAT(3I3)
05128 IF(NE,NE,0) NE=NC
05130 WRITE(6,3) NC, NPR, NE
05132 3 FORMAT(16H1 NC NPR NE,/,3I5,/)
05134 IF(NC,EQ,0) RETURN
05136 READ(5,2) XC, YC
05138 2 FORMAT(10F8,4)
05140 READ(5,2) (ZC(I), I=1, NC)
05142 READ(5,2) (RHOC(I), I=1, NPR)
05144 IF(NE,NE,0) READ(5,2) (ZE(I), I=1, NE)
05146 WRITE(6,4)
05148 4 FORMAT(37H CAP XC YC ZC ZE,/)
05150 DO 100 I=1, NC
05152 IF(NE,EQ,0) WRITE(6,5) I, XC, YC, ZC(I)
05154 IF(NE,NE,0) WRITE(6,5) I, XC, YC, ZC(I), ZE(I)
05156 100 CONTINUE
05158 5 FORMAT(14,4X,4F8,4)
05160 WRITE(6,6)
05162 6 FORMAT(/,5H RHOC)
05164 WRITE(6,7) (RHOC(I), I=1, NPR)
05166 7 FORMAT(1X,10F8,4)
05168 IF(MOD(NPR,2),NE,1) GO TO 980
05170 DO 120 I=2, NPR
05172 IF(RHOC(I),LE,RHOC(I-1)) GO TO 980
05174 120 COMPUTATION OF CAP SECTOR PARAMETERS.
05176 DO 47 I=1, NP
05178 RC(I)=SQRT((YB(I)-YC)**2+(XB(I)-XC)**2)
05180 47 CONTINUE
05182 DO 57 I=2, NP
05184 I2=I-1
05186 RR1=YB(I2)-YC
05188 RR2=XB(I2)-XC
05190 RC1(I2)=SQRT(RR1*RR1+RR2*RR2)
05192 AC(I2)=ABS((XB(I)-XC)*(YB(I2)+YC)+(XB(I2)-XB(I))*
05194 1 (YB(I2)+YB(I))+(XC-XB(I2))*(YC+YB(I2)))/2.
05196 SPC(I2)=RR1/RC1(I2)
05198 CPC(I2)=RR2/RC1(I2)
05200 57 CONTINUE
05202 DO 67 I=2, NPR
05204 I2=I-1
05206 RHOC1(I2)=(RHOC(I)+RHOC(I2))/2.
05208 DRHOC(I2)=RHOC(I)-RHOC(I2)
05210 67 CONTINUE
05212 67 COMPUTATION OF CAP TRIANGLE FUNCTIONS.
05214 LC=(NPR-3)/2
05216 DO 74 J=1, LC

```

```

05212 J2=2*(J-1)+1
05214 J3=J2+1
05216 J4=J3+1
05218 J5=J4+1
05220 J6=4*(J-1)+1
05222 J7=J6+1
05224 J8=J7+1
05226 J9=J8+1
05228 DEL1=DRHOC(J2)+DRHOC(J3)
05230 DEL2=DRHOC(J4)+DRHOC(J5)
05232 TPCR(J6)=1./DEL1
05234 TPCR(J7)=1./DEL1
05236 TPCR(J8)=-1./DEL2
05238 TPCR(J9)=-1./DEL2
05240 TCR(J6)=DRHOC(J2)/2./DEL1
05242 TCR(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
05244 TCR(J8)=(DRHOC(J4)/2.+DRHOC(J5))/DEL2
05246 TCR(J9)=DRHOC(J5)/2./DEL2
05248 74 CONTINUE
05250 LC4=LC*4
05252 DD 75 I=1,LC4
05254 TCT(I)=TCR(I)
05256 TPCT(I)=TPCR(I)
05258 75 CONTINUE
05260 TCT(LC4-1)=2.-TCT(LC4-1)
05262 TCT(LC4)=2.-TCT(LC4)
05264 TPCT(LC4-1)=-TPCT(LC4-1)
05266 TPCT(LC4)=-TPCT(LC4)
05268 IF(RHOC(I).EQ.0.0) GO TO 76
05270 TCT(1)=2.-TCT(1)
05272 TCT(2)=2.-TCT(2)
05274 TPCT(1)=-TPCT(1)
05276 TPCT(2)=-TPCT(2)
05278 76 CONTINUE
05280 IF(NE.EQ.0) RETURN
05282 COMPUTATION OF EDGE HALF TRIANGLE FUNCTIONS.
05284 DO 80 IC=1,NC
05286 J2=NCR-2
05288 J3=J2+1
05290 J6=2*(IC-1)+1
05292 J7=J6+1
05294 DEL1=DRHOC(J2)+DRHOC(J3)
05296 TPCR(J6)=1./DEL1
05298 TPCR(J7)=1./DEL1
05300 TCE(J6)=DRHOC(J2)/2./DEL1
05302 TCE(J7)=(DRHOC(J2)+DRHOC(J3)/2.)/DEL1
05304 DEL2=ZC(IC)-ZC(IC)
05306 ZBE(J6)=ZC(IC)+0.25*DEL2
05308 ZBE(J7)=ZC(IC)+0.75*DEL2
05310 IF(DEL2.LT.0.0) GO TO 78
05312 78 IS AT 2=-1.
    TPBE(J6)=-1./DEL2

```

```

05314      TPBE(J7)=-1./DEL2
05316      TBE(J6)=-0.75
05318      TBE(J7)=-0.25
05320      GO TO 80
05322      CONTINUE
05324C 78  EDGE IS AT Z=L.
05326      TPBE(J6)=-1./DEL2
05328      TPBE(J7)=-1./DEL2
05330      TBE(J6)=-0.75
05332      TBE(J7)=-0.25
05334      80  CONTINUE
05336      RETURN
05338 980  WRITE(6,981)
05340 981  FORMAT(/, ' **** ERROR IN CAP INPUT')
05342      STOP
05344      END
-----
05310C  SUBROUTINE WIREIN
05315
05320C  READ WIRE COORDINATES AND COMPUTE WIRE SEGMENT ARRAYS.
05325C
05330C
05335      COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
05340 1 XW(100),YW(100),ZW(100)
05345      COMMON /WIRE2/ DWW(100),DXW(100),DYW(100),DZW(100)
05350      COMMON /WIRE3/ NW,INDW(6),RADW(5)
05355      COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
05360      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
05365      COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
05370      COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
05375      COMMON /JUNC4/ UXJ(10),UYJ(10),UZJ(10)
05380      COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
05385      COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
05390      NW=0
05395      NJ=0
05400      LW=0
05405      READ(5,51) NW,NPW,NJ
05410 51  FORMAT(3I3)
05415      WRITE(6,1) NW,NPW,NJ
05420 1  FORMAT(24H1 NW NPW NJ,/,3I6)
05425      IF(NW.EQ.0) RETURN
05430      READ(5,53) (XW(I),I=1,NPW)
05435      READ(5,53) (YW(I),I=1,NPW)
05440      READ(5,53) (ZW(I),I=1,NPW)
05445 53  FORMAT(10F8.4)
05450      READ(5,52) (INDW(I),I=1,NW)
05455 52  FORMAT(10I8)
05460      INDW(NW+1)=NPW+1
05465      READ(5,53) (RADW(I),I=1,NW)
05470      IF(NJ.EQ.0) GO TO 50
05475C NOTE, INDJW MUST BE MONOTONIC INCREASING.
05480      READ(5,52) (INDJW(I),I=1,NJ)
05485      READ(5,53) (RADD(I),I=1,NJ)

```

```

05690      READ(5,53) (UXJ(I),I=1,NJ)
05695      READ(5,53) (UYJ(I),I=1,NJ)
05700      READ(5,53) (UZJ(I),I=1,NJ)
05705 50    IERN=0
05710      IERJ=C
05715      WRITE(6,61)
05720 61    FORMAT(/,22X,'WIRE COORDINATES',20X,'JUNCTION PARAMETERS',/,
05725      1 26X,'IW',4X,'XW',6X,'YW',6X,'ZW',
05730      2 10X,'IJ',4X,'RADD',5X,'UXJ',5X,'UYJ',5X,'UZJ')
05735      IJ=1
05740C THIS LOOP LISTS WIRE/JUNCTION POINTS, WHILE CHECKING THE FOLLOWING:
05745C 1) EACH WIRE MUST CONTAIN AN ODD NUMBER OF POINTS.
05750C 2) EACH JUNCTION MUST EITHER START OR TERMINATE A WIRE.
05755C 3) CHECK THAT ALL JUNCTION POINTS ARE FOUND.
05760      DO 100 IW=1,NW
05765      WRITE(6,62) IW,RADW(IW)
05770 62    FORMAT(2X,'WIRE',I3,' RADW=',F8.4)
05775      I1=INDW(IW)
05780      I2=INDW(IW+1)-1
05785C CHECK FOR AN ODD NUMBER OF POINTS ON WIRE IW.
05790      IF(MOD(I2-I1+1,2).NE.1) IERN=1
05795      DO 90 I=I1,I2
05800      WRITE(6,63) I,XW(I),YW(I),ZW(I)
05805 63    FORMAT(25X,I3,3F8.4)
05810C CHECK IF WIRE POINT I IS A JUNCTION POINT.
05815      IF(IJ.GT.NJ) GO TO 90
05820      IF(INDJW(IJ).NE.I) GO TO 90
05825C CHECK THAT JUNCTION POINT IJ IS AT THE START OR END OF WIRE.
05830      IF(I.NE.I1.AND. I.NE.I2) IERJ=1
05835      WRITE(6,64) IJ,RADD(IJ),UXJ(IJ),UYJ(IJ),UZJ(IJ)
05840 64    FORMAT(14,58X,I3,4F8.4)
05845C DETERMINE DIRECTION IN WHICH WIRE LEAVES JUNCTION POINT.
05850      INDJ(IJ)=1
05855      IF(I.EQ.I2) INDJ(IJ)=-1
05860C COMPUTE JUNCTION PARAMETERS.
05865      RADJ(IJ)=RADW(IW)
05870      XJ(IJ)=XW(I)
05875      YJ(IJ)=YW(I)
05880      ZJ(IJ)=ZW(I)
05885      IJ=IJ+1
05890 90    CONTINUE
05895 100   CONTINUE
05900C CHECK FOR WIRE OR JUNCTION INPUT ERRORS.
05905      IF(IERN.NE.0) GO TO 980
05910      IF(IJ-1.NE.NJ.OR. IERJ.NE.0) GO TO 990
05915C COMPUTATION OF WIRE SEGMENT PARAMETERS.
05920      DO 57 I=2,NPW
05925      I2=I-1
05930      DXW(I2)=XW(I)-XW(I2)
05935      DYW(I2)=YW(I)-YW(I2)
05940      DZW(I2)=ZW(I)-ZW(I2)
05945      DHW(I2)=SQRT(DXW(I2)**2+DYW(I2)**2+DZW(I2)**2)

```

```

03950      XW1(I2)=0.5*(XW(I)+XW(I2))
03955      YW1(I2)=0.5*(YW(I)+YW(I2))
03960      ZW1(I2)=0.5*(ZW(I)+ZW(I2))
03965      57 CONTINUE
03970C COMPUTATION OF WIRE TRIANGLE FUNCTIONS TW.
03975      LW=0
03980      DO 75 IW=1,NW
03985      I1=INDW(IW)
03990      I2=INDW(IW+1)-1
03995      LW1=(I2-I1-2)/2
06000      DO 74 J=1,LW1
06005      I3=I1+1
06010      J2=I1+(J-1)+I1
06015      J3=J2+1
06020      J4=J3+1
06025      J5=J4+1
06030      J6=4*(LW-1)+1
06035      J7=J6+1
06040      J8=J7+1
06045      J9=J8+1
06050      INDTW(LW)=J2
06055      DEL1=DHW(J2)+DHW(J3)
06060      DEL2=DHW(J4)+DHW(J5)
06065      TPW(J6)=1./DEL1
06070      TPW(J7)=1./DEL1
06075      TPW(J8)=-1./DEL2
06080      TPW(J9)=-1./DEL2
06085      TW(J6)=DHW(J2)/2./DEL1
06090      TW(J7)=(DHW(J2)+DHW(J3))/2./DEL1
06095      TW(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
06100      TW(J9)=DHW(J5)/2./DEL2
06105      74 CONTINUE
06110      75 CONTINUE
06115C NOTE, LW=(NPW-NW)/2-NW
06120      IF(NJ.EQ.0) RETURN
06125C COMPUTATION OF JUNCTION HALF TRIANGLE FUNCTIONS TJ.
06130      DO 85 IJ=1,NJ
06135      IF(INDJ(IJ).GT.0) GO TO 80
06140C JUNCTION IS AT THE END OF A WIRE.
06145      J2=INDJW(IJ)-2
06150      J3=J2+1
06155      J6=2*(IJ-1)+1
06160      J7=J6+1
06165      INDJ(IJ)=J2
06170      DEL1=-DHW(J2)-DHW(J3)
06175      TJ(J6)=1./DEL1
06180      TJ(J7)=1./DEL1
06185      TJ(J6)=DHW(J2)/2./DEL1
06190      TJ(J7)=(DHW(J2)+DHW(J3))/2./DEL1
06195      GO TO 85
06200C JUNCTION IS AT THE START OF A WIRE.
06205      80      J4=INDJW(IJ)

```

```

06210      J5=J4+1
06215      J8=2*(IJ-1)+1
06220      J9=J8+1
06225      INDJ(IJ)=J4
06230      DEL2=DHW(J4)+DHW(J5)
06235      TPJ(J8)=-1./DEL2
06240      TPJ(J9)=-1./DEL2
06245      TJ(J8)=(DHW(J4)/2.+DHW(J5))/DEL2
06250      TJ(J9)=DHW(J5)/2./DEL2
06255      85 CONTINUE
06260C COMPUTE UNIT VECTORS FOR JUNCTION(DISK).
06265      DO 95 IJ=1,NJ
06270C NORMAL UNIT VECTOR (UJ).
06275      RR=SQRT(UXJ(IJ)**2+UYJ(IJ)**2+UZJ(IJ)**2)
06280      UXJ(IJ)=UXJ(IJ)/RR
06285      UYJ(IJ)=UYJ(IJ)/RR
06290      UZJ(IJ)=UZJ(IJ)/RR
06295C FIND 2 ORTHOGONAL UNIT VECTORS IN THE PLANE OF THE DISK (UJ1 & UJ2).
06300      UXJ1(IJ)=0.0
06305      UYJ1(IJ)=0.0
06310      UZJ1(IJ)=0.0
06315      UXJ2(IJ)=0.0
06320      UYJ2(IJ)=0.0
06325      UZJ2(IJ)=0.0
06330      IF(UXJ(IJ).EQ.0.0) GO TO 91
06335C FIND INTERSECTION WITH X-Y PLANE.
06340      UYJ1(IJ)=1.0
06345      UXJ1(IJ)=-UYJ(IJ)/UXJ(IJ)
06350C FIND INTERSECTION WITH X-Z PLANE.
06355      UZJ2(IJ)=1.0
06360      UXJ2(IJ)=-UZJ(IJ)/UXJ(IJ)
06365      GO TO 94
06370 91 IF(UYJ(IJ).EQ.0.0) GO TO 92
06375C FIND INTERSECTION WITH Y-Z PLANE.
06380      UZJ1(IJ)=1.0
06385      UYJ1(IJ)=-UZJ(IJ)/UYJ(IJ)
06390C FIND INTERSECTION WITH X-Y PLANE.
06395      UXJ2(IJ)=1.0
06400      UYJ2(IJ)=-UXJ(IJ)/UYJ(IJ)
06405      GO TO 94
06410 92 IF(UZJ(IJ).EQ.0.0) GO TO 94
06415C FIND INTERSECTION WITH X-Z PLANE.
06420      UXJ1(IJ)=1.0
06425      UZJ1(IJ)=-UXJ(IJ)/UZJ(IJ)
06430C FIND INTERSECTION WITH Y-Z PLANE.
06435      UYJ2(IJ)=1.0
06440      UZJ2(IJ)=-UYJ(IJ)/UZJ(IJ)
06445      94 CONTINUE
06450      RR=SQRT(UXJ1(IJ)**2+UYJ1(IJ)**2+UZJ1(IJ)**2)
06455      UXJ1(IJ)=UXJ1(IJ)/RR
06460      UYJ1(IJ)=UYJ1(IJ)/RR
06465      UZJ1(IJ)=UZJ1(IJ)/RR

```

```

06470      RR=SQRT(UXJ2(IJ)**2+UYJ2(IJ)**2+UZJ2(IJ)**2)
06475      UXJ2(IJ)=UXJ2(IJ)/RR
06480      UYJ2(IJ)=UYJ2(IJ)/RR
06485      UZJ2(IJ)=UZJ2(IJ)/RR
06490 95    CONTINUE
06495      RETURN
06500 980   WRITE(6,981)
06505 981   FORMAT(//, ' **** ERROR IN WIRE INPUT')
06510      STOP
06515 990   WRITE(6,991)
06520 991   FORMAT(//, ' **** ERROR IN JUNCTION INPUT')
06525      STOP
06530      END
07940C -----
07950      SUBROUTINE ORDER(MORD)
07960C DETERMINE THE ORDER IN WHICH THE Y MATRIX WAS GENERATED, AND
07970C CHECK FOR DIMENSIONAL CONSISTENCY BETWEEN THE Y MATRIX DATA FILE,
07980C AND THE INPUT DATA FILE.
07990      DIMENSION MORD(2)
08000      COMMON /BOT1/ NMODE,NPT,NBAND
08010      COMMON /BOT2/ NP,8L,Y8(83),XB(83),Y81(82),XB1(82)
08020      COMMON /CAP1/ NC,XC,YC,ZC(2)
08025      COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
08030      COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
08032 1 XW1(100),YW1(100),ZW1(100)
08034      COMMON /WIRE3/ NW,INOW(6),RADW(5)
08040      COMMON /JUNC1/ NJ,INOW(10),RADJ(10),RADD(10)
08042      COMMON /EDG1/ NE,ZE(2),ZBE(10)
08050      READ(1) NMODE1,NP1,NC1,NPR1,NE1,NW1,NPW1,NJ1
08052      IF(NMODE.NE.NMODE1) GO TO 500
08054      IF(NP.NE.NP1) GO TO 500
08056      IF(NC.NE.NC1) GO TO 500
08058      IF(NPR.NE.NPR1) GO TO 500
08060      IF(NE.NE.NE1) GO TO 500
08062      IF(NW.NE.NW1) GO TO 500
08064      IF(NPW.NE.NPW1) GO TO 500
08066      IF(NJ.NE.NJ1) GO TO 500
08120C MORD SPECIFIES THE ORDER IN WHICH CAPS AND/OR WIRES HAVE BEEN ADDED
08130C TO THE BOT. INDEX 1 REFERS TO CAPS AND INDEX 2 REFERS TO WIRES.
08140C MORD(1)=0 IF 1 HAS NOT BEEN ADDED
08150C MORD(1)=N IF 1 HAS BEEN ADDED TO THE SYSTEM MATRIX, AND
08160C IS LOCATED AT PSEUDO MODE NUMBER N.
08170      READ(1) MORD
08172      WRITE(6,4) MORD
08220 4      FORMAT(1, ' THE Y MATRIX CONTAINS THE FOLLOWING ADDITIONS ( 0 IF',
08230 1 ' NOT INCLUDED, OR CORRESPONDING MODE NUMBER IF PRESENT)',//,
08240 2 10X, 'CAPS',I4,/,1CX, 'WIRES',I3)
08250      RETURN
08270 500   WRITE(6,1)
08280 1      FORMAT(//, ' *** ERROR *** INPUT PARAMETERS DO NOT CHECK WITH',
08290 1 ' THE PARAMETERS IN THE Y MATRIX FILE',//,
08300 2 6X, ' NMODE NP NC NPR NE NW NPW NJ')

```

```

08310      WRITE(6,2) NMODE,NP,NC,NPR,NE,NW,NPW,NJ
08320 2      FORMAT(' INPUT ',8I6)
08330      WRITE(6,3) NMODE1,NP1,NC1,NPR1,NE1,NW1,NPW1,NJ1
08340 3      FORMAT(' Y FILE ',8I6)
08350      STOP
08360      END
08670C *****
08680      SUBROUTINE RBT(RBT,RBP,NT,TH,PHI)
08690C *****
08700C COMPUTE BOT TRANSFER MATRICES FOR THETA AND PHI POLARIZATION.
08710      COMPLEX RBT(1),RBP(1),A6,U
08720      COMMON /WAVE/ BK
08730      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
08740      COMMON /BOT3/ DH(82),SV(82),CV(82)
08750      COMMON /BOT5/ T(160),TP(160),TZ(160)
08760      DIMENSION TH(1),PHI(1)
08770      DATA PI,U /3.14159265,(0.,1.)/
08780      DATA DTOR /0.017453292/
08790      LS=NP-3
08800      NM=LS/2
08810      DO 156 K=1,NT
08820      K1=(K-1)*LS
08830      CS=COS(TH(K)*DTOR)
08840      SN=SIN(TH(K)*DTOR)
08850      CSPHI=COS(PHI(K)*DTOR)
08860      SNPHI=SIN(PHI(K)*DTOR)
08870      DO 300 J=1,NM
08880      J1=J+K1
08890      J2=J1+NM
08900      RBT(J1)=0.
08910      RBT(J2)=0.
08920      RBP(J1)=0.
08930      RBP(J2)=0.
08940C CALCULATION OF TRANSFER MATRICES (EQS. 38-41).
08950C NOTE, ONLY THE MODE INDEPENDENT PORTION IS COMPUTED HERE.
08960      DO 301 I=1,4
08970      I1=2*(J-1)+1
08980      I4=4*(J-1)+1
08990      A6=CEXP(U*BK+SN*(XB1(I1)*CSPHI+YB1(I1)*SNPHI))
09000C THETA POLARIZED (EQS. 38-39).
09010      RBT(J1)=RBT(J1)+DH(I1)*T(I4)*(CV(I1)*CSPHI+SV(I1)*SNPHI)*A6
09020      RBT(J2)=RBT(J2)+DH(I1)*TZ(I4)*A6
09030C PHI POLARIZED (EQS. 40-41).
09040      RBP(J1)=RBP(J1)+DH(I1)*T(I4)*(SV(I1)*CSPHI-CV(I1)*SNPHI)*A6
09050 301 CONTINUE
09060      RBT(J1)=CS*RBT(J1)
09070      RBT(J2)=-SN*RBT(J2)
09080 300 CONTINUE
09090 156 CONTINUE
09100      RETURN
09110      END
09111C -----

```

```

09112      SUBROUTINE RCAP(RCT,RCP,NT,TH,PHI)
09113C
09114C      COMPUTE CAP TRANSFER MATRICES FOR THETA AND PHI POLARIZATION.
09115C
09116      COMPLEX RCT(1),RCP(1),A6,U
09117      COMMON /WAVE/ BK
09118      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
09119      COMMON /BOT3/ DH(82),SV(82),CV(82)
09120      COMMON /BOT5/ T(160),TP(160),TZ(160)
09121      COMMON /CAP1/ NC,XC,YC,ZC(2)
09122      COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
09123      COMMON /CAP3/ TCR(36),TCT(36),TPCR(36),TPCT(36)
09124      COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
09125      COMMON /EDG1/ NE,ZE(2),ZOE(10)
09126      COMMON /EDG2/ TCE(10),TPCE(10),TBE(10),TPBE(10)
09127      DIMENSION TH(1),PHI(1)
09128      DATA PI,U /3.14159265,(0.,1.)/
09129      DATA DTOR /0.017453292/
09130      NM=(NP-3)/2
09131      LR=(NPR-3)/2
09132      LC=NC+NM+LR
09133      LE=NE+NM
09134      DO 156 K=1,NT
09135      K1=(K-1)*(2*LC+LE)
09136      CS=CCS(TH(K)*DTOR)
09137      SN=SIN(TH(K)*DTOR)
09138      CSPHI=CCS(PHI(K)*DTOR)
09139      SNPHI=SIN(PHI(K)*DTOR)
09140      J1=K1
09141      DO 300 JC=1,NC
09142      DO 300 JP=1,NM
09143      DO 300 JR=1,LR
09144      J1=J1+1
09145      J2=J1+LC
09146      RCT(J1)=0.
09147      RCT(J2)=0.
09148      RCP(J1)=0.
09149      RCP(J2)=0.
09150C      CALCULATION OF TRANSFER MATRICES.
09151      DO 301 IIP=1,4
09152      IP1=2*(JP-1)+IIP
09153      IP4=4*(JP-1)+IIP
09154      DO 301 IIR=1,4
09155      IR1=2*(JR-1)+IIR
09156      IR4=4*(JR-1)+IIR
09157      XX=XC+RHOC1(IR1)*RC1(IP1)*CPC(IP1)
09158      YY=YC+RHOC1(IR1)*RC1(IP1)*SPC(IP1)
09159      ZZ=ZC(JC)
09160      A6=CEXP(U*BK*(SN*(XX*CSPHI+YY*SNPHI)+ZZ*CS))
09161      AI=AC(IP1)*ABS(RHOC(IR1+1)*2-RHOC(IR1)*2)
09162C      THETA POLARIZED.
09163      RCT(J1)=RCT(J1)+AI*(CV(IP1)*CSPHI+SV(IP1)*SNPHI)*

```

```

09164      1 T(IP4)*TCT(IR4)*A6
09165      RCT(J2)=RCT(J2)+AI*(CPC(IP1)*CSPHI+SPC(IP1)*SNPHI)*
09166      1 TZ(IP4)*TCR(IR4)*A6
09167C PHI POLARIZED.
09168      RCP(J1)=RCP(J1)+AI*(SV(IP1)*CSPHI-CV(IP1)*SNPHI)*
09169      1 T(IP4)*TCT(IR4)*A6
09170      RCP(J2)=RCP(J2)+AI*(SPC(IP1)*CSPHI-CPC(IP1)*SNPHI)*
09171      1 TZ(IP4)*TCR(IR4)*A6
09172      301 CONTINUE
09173      RCT(J1)=CS*RCT(J1)
09174      RCT(J2)=CS*RCT(J2)
09175      300 CONTINUE
09176      IF(NE.EQ.0) GO TO 156
09177C EDGE TRANSFER MATRICES FOLLOW.
09178      J2=2*(LC+K1)
09179      DO 400 JC=1,NC
09180      DO 400 JP=1,NM
09181      J2=J2+1
09182      RCT(J2)=0.
09183      RCP(J2)=0.
09184      DO 401 IIP=1,4
09185      IP1=2*(JP-1)+IIP
09186      IP4=4*(JP-1)+IIP
09187      DO 401 IIRZ=1,2
09188      IR1=2*(LR+IIRZ)
09189      IR4=2*(JC-1)+IIRZ
09190      IZ1=2*(JC-1)+IIRZ
09191      IZ4=2*(JC-1)+IIRZ
09192      XX=XC+RHOC1(IR1)*RC1(IP1)*CPC(IP1)
09193      YY=YC+RHOC1(IR1)*RC1(IP1)*SPC(IP1)
09194      ZZ=ZC(JC)
09195      A6=CEXP(U*BX*(SN*(XX*CSPHI+YY*SNPHI)+ZZ*CS))
09196      AI=AC(IP1)*ABS(RHOC(IR1+1))*2-RHOC(IR1)*2)
09197C EDGE (CAP)
09198C THETA POLARIZED.
09199      RCT(J2)=RCT(J2)+CS*AI*(CPC(IP1)*CSPHI+SPC(IP1)*SNPHI)*
09200      1 TZ(IP4)*TCE(IR4)*A6
09201C PHI POLARIZED.
09202      RCP(J2)=RCP(J2)+AI*(SPC(IP1)*CSPHI-CPC(IP1)*SNPHI)*
09203      1 TZ(IP4)*TCE(IR4)*A6
09204      XX=XB1(IP1)
09205      YY=YB1(IP1)
09206      ZZ=ZB1(IZ1)
09207      A6=CEXP(U*BX*(SN*(XX*CSPHI+YY*SNPHI)+ZZ*CS))
09208      AI=DH(IP1)*ABS(ZE(JC)-ZC(JC))/2.
09209C EDGE (BOT)
09210C THETA POLARIZED.
09211      RCT(J2)=RCT(J2)-SN*AI*TZ(IP4)*TBE(IZ4)*A6
09212      401 CONTINUE
09213      400 CONTINUE
09214      156 CONTINUE
09215      RETURN

```

```

09216      END
09217C -----
09218      SUBROUTINE RWIRE(RWT,RWP,NT,TH,PHI)
09219C -----
09220C      COMPUTE WIRE-JUNCTION TRANSFER MATRICES.
09230C
09240      COMPLEX RWT(1),RWP(1),A6,U
09245      COMMON /WAVE/ BK
09250      COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
09260      1 XW1(100),YW1(100),ZW1(100)
09270      COMMON /WIRE2/ DHW(100),DXW(100),DYW(100),DZW(100)
09280      COMMON /WIRE3/ NW,INDW(6),RADW(5)
09290      COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
09300      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
09310      COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
09320      COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
09324      COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
09326      COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
09340      DIMENSION TH(1),PHI(1)
09350      DATA PI,U /3.14159265,(0.,1.) /
09360      DATA DTOR /0.017453292/
09380      LWJ=LW+NJ
09390      DO 156 K=1,NT
09400      K1=(K-1)*LWJ
09410      CS=CCS(TH(K)*DTOR)
09420      SN=SIN(TH(K)*DTOR)
09430      CSPHI=CCS(PHI(K)*DTOR)
09440      SNPHI=SIN(PHI(K)*DTOR)
09450      DO 300 J=1,LW
09455      J1=J+K1
09460      RWT(J1)=0.
09465      RWP(J1)=C.
09470      DO 301 I=1,4
09475      I1=INDTW(J)+I-1
09480      I4=4*(J-1)+I
09485      A6=CEXP(U*BK*(SN*(XW1(I1)*CSPHI+YW1(I1)*SNPHI)+ZW1(I1)*CS))
09490      RWT(J1)=RWT(J1)+(CS*(DXW(I1)*CSPHI+DYW(I1)*SNPHI)-
09495      1 DZW(I1)*SN)*TW(I4)*A6
09500      RWP(J1)=RWP(J1)+(-DXW(I1)*SNPHI+DYW(I1)*CSPHI)*TW(I4)*A6
09505      301 CONTINUE
09510      300 CONTINUE
09515      IF(NJ.EQ.0) GO TO 156
09520      J1=LW+K1
09525      DO 400 J=1,NJ
09530      J1=J1+1
09535      RWT(J1)=0.
09540      RWP(J1)=C.
09545      DO 401 I=1,2
09550      I1=INDTJ(J)+I-1
09555      I2=2*(J-1)+I
09560      A6=CEXP(U*BK*(SN*(XW1(I1)*CSPHI+YW1(I1)*SNPHI)+ZW1(I1)*CS))
09565      RWT(J1)=RWT(J1)+(CS*(DXW(I1)*CSPHI+DYW(I1)*SNPHI)-

```

```

09570      1 DZW(I1)*SN)*TJ(I2)*A6
09575      RWP(J1)=RWP(J1)+(-DXW(I1)*SNPHI+DYW(I1)*CSPHI)*TJ(I2)*A6
09580 401 CONTINUE
09585      DO 402 JJ=1,4
09590      ALPHA=PI*((JJ-1)/2.+0.25)
09595      SA=SIN(ALPHA)
09600      CA=COS(ALPHA)
09605      RA=(RADJ(J)+RADD(J))/2.
09610      XA=XJ(J)+RA*(CA*UXJ1(J)+SA*UXJ2(J))
09612      YA=YJ(J)+RA*(CA*UYJ1(J)+SA*UYJ2(J))
09614      ZA=ZJ(J)+RA*(CA*UZJ1(J)+SA*UZJ2(J))
09616      DT=CA*(UXJ1(J)*CS*CSPHI+UYJ1(J)*CS*SNPHI-UZJ1(J)*SN)+
09618      1 SA*(UXJ2(J)*CS*CSPHI+UYJ2(J)*CS*SNPHI-UZJ2(J)*SN)
09620      DP=CA*(-UXJ1(J)*SNPHI+UYJ1(J)*CSPHI)+
09622      1 SA*(-UXJ2(J)*SNPHI+UYJ2(J)*CSPHI)
09625      RR=(RADJ(J)-RADD(J))/2.
09630      A6=CEXP(U*BK*(SN*(XA*CSPHI+YA*SNPHI)+ZA*CS))
09635      RWT(J1)=RWT(J1)+RR/4.*DT*A6
09645      RWP(J1)=RWP(J1)+RR/4.*DP*A6
09650 402 CONTINUE
09660 400 CONTINUE
09670 156 CONTINUE
09680      RETURN
09690      END
09700C *****
09702C SUBROUTINE LPCUR(NMODE,NM,CB,LW,NJ,CW,LC,LE,CC)
09704C *****
09706C LIST AND PLOT BOT, WIRE, AND CAP CURRENTS.
09708C COMPLEX U,A3,A4,CB(1),CW(1),CC(1)
09710C DIMENSION TMAG(41),TPHS(41),ZMAG(41),ZPHS(41)
09712C DATA PI,U /3.14159265,(0.,1.)/
09714C WRITE(6,3)
09716 3 FORMAT(16H1 BOT CURRENTS)
09718C KM=2*NMODE-1
09720C LS=NM*2
09722C DO 100 NM=1,KM
09724C M=NM-NMODE
09726C I1=(NM-1)*LS+1
09728C I2=I1+NM-1
09730C WRITE(6,1) M,(CB(I),I=I1,I2)
09732 1 FORMAT(/,29H Y-DIRECTED CURRENTS FOR MODE,I3,/,4(2E12.4,5X))
09734C I1=I2+1
09736C I2=I1+NM-1
09738C WRITE(6,2) M,(CB(I),I=I1,I2)
09740 2 FORMAT(/,29H Z-DIRECTED CURRENTS FOR MODE,I3,/,4(2E12.4,5X))
09742 100 CONTINUE
09744C DO 320 I=1,NM
09746C DO 310 K=1,41
09748C A3=0.0
09750C A4=0.0
09752C ZE=-1.0+(K-1)*0.05
09754C M=-NMODE

```

```

09756      I1=I-LS
09758      DO 3C5 MM=1,KM
09760      M=M+1
09762      I1=I+LS
09764      I2=I+NM
09766C SUM THE T AND Z DIRECTED CURRENTS ON TRIANGLE FUNCTION I AT Z=ZE,
09768C (EQ. 8).
09770      A3=A3+CB(I1)*CEXP(U*M*PI*ZE)
09772      A4=A4+CB(I2)*CEXP(U*M*PI*ZE)-((-1)**M)
09774 305 CONTINUE
09776      TMAG(K)=CABS(A3)/2.
09778      TPHS(K)=ATAN2(AIMAG(A3),REAL(A3))*180./PI
09780      ZMAG(K)=CABS(A4)/2.
09782      ZPHS(K)=ATAN2(AIMAG(A4),REAL(A4))*180./PI
09784 31C CONTINUE
09786      WRITE(6,301)
09788 3C1 FORMAT(1H1)
09790      WRITE(6,302) I
09792 302 FORMAT(41H T-DIRECTED CURRENTS ON TRIANGLE FUNCTION,I3,/)
09794C PLOT MAGNITUDE AND PHASE.
09796      CALL PLOTG(TMAG,TPHS)
09798      WRITE(6,304) I
09800 304 FORMAT(///,41H Z-DIRECTED CURRENTS ON TRIANGLE FUNCTION,I3,/)
09802C PLOT MAGNITUDE AND PHASE.
09804      CALL PLOTG(ZMAG,ZPHS)
09806 320 CONTINUE
09808      IF(LW.EQ.0) GO TO 500
09810      WRITE(6,4)
09812 4      FORMAT(//,16H WIRE CURRENTS,/)
09814      WRITE(6,5) (CW(I),I=1,LW)
09816 5      FORMAT(4(2E12.4,5X))
09818 500 CONTINUE
09820      IF(NJ.EQ.0) GO TO 600
09822      I1=LW+1
09824      I2=LW+NJ
09826      WRITE(6,6)
09828 6      FORMAT(//,20H JUNCTION CURRENTS,/)
09830      WRITE(6,5) (CW(I),I=I1,I2)
09832 600 CONTINUE
09834      IF(LC.EQ.0) RETURN
09836      WRITE(6,7)
09838 7      FORMAT(//,15H CAP CURRENTS,/)
09840      WRITE(6,8)
09842 8      FORMAT(//,11H T-DIRECTED)
09844      WRITE(6,5) (CC(I),I=1,LC)
09846      WRITE(6,9)
09848 9      FORMAT(//,11H R-DIRECTED)
09850      I1=LC+1
09852      I2=2*LC
09854      WRITE(6,5) (CC(I),I=I1,I2)
09856      IF(LE.EQ.0) RETURN
09858      WRITE(6,10)

```

```

09860 10  FORMAT(/,14H EDGE CURRENTS)
09862      I1=2*LC+1
09864      I2=2*LC+LE
09866      WRITE(6,5) (CC(I),I=I1,I2)
09868      RETURN
09870      END
10270C *****
10280      SUBROUTINE PLOT(Y1,Y2)
10290C *****
10300C
10310C WRITTEN 2/14/74 BY J. M. PUTNAM DEPT 220 X23877
10320C
10330C THIS ROUTINE PRODUCES A LINEAR XY PLOT.
10340C N IS THE NUMBER OF POINTS TO BE PLOTTED.
10350C NR IS THE NUMBER OF ROWS TO BE USED FOR THE Y-AXIS.
10360C NC IS THE NUMBER OF COLUMNS TO BE USED FOR THE X-AXIS.
10370C NOTE, NC-1 MUST BE DIVISIBLE BY 10 AND LESS THAN 102.
10380C
10390      REAL Y1(41),Y2(41),HEAD(10)
10400      INTEGER LINE(101),BLANK,STAR,PLUS
10410      DATA BLANK,STAR,PLUS /1H,1H*,1H+/
10420      N=41
10430      NR=21
10440      NC=41
10450      N10=(NC-1)/10
10460      WRITE(6,500)
10470 500  FORMAT(12H * MAGNITUDE,45X,7H+ PHASE)
10480      WRITE(6,50+)
10490      XMIN=-1.0
10500      XMAX=1.0
10510      YMIN1=Y1(1)
10520      YMAX1=Y1(1)
10530      YMIN2=-180.0
10540      YMAX2=180.0
10550      DO 6 I=1,N
10560      IF(Y1(I).LT.YMIN1) YMIN1=Y1(I)
10570      IF(Y1(I).GT.YMAX1) YMAX1=Y1(I)
10580 6      CONTINUE
10590      DO 5 I=1,N10
10600      Z=I
10610 5      HEAD(I)=(XMAX-XMIN)*Z/N10+XMIN
10620      DY1=(YMAX1-YMIN1)/(NR-1)
10630      DY2=18.0
10640      Z1=YMAX1+DY1
10650      Z2=YMAX2+DY2
10660      YL1=Z1-DY1/2.
10670      YL2=Z2-DY2/2.
10680      DO 7 J=1,NR
10690      DO 8 K=1,NC
10700 8      LINE(K)=BLANK
10710      Z1=Z1-DY1
10720      Z2=Z2-DY2

```

```

10730      YU1=YL1
10740      YU2=YL2
10750      YL1=Z1-DY1/2.
10760      YL2=Z2-DY2/2.
10770      DO 9 I=1,N
10780      X=XMIN+(XMAX-XMIN)*(I-1)/FLOAT(N-1)
10800      IF(Y2(I).GE.YU2) GO TO 10
10810      IF(Y2(I).LT.YL2) GO TO 10
10820      K=(X-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
10830      IF(K.GT.NC) K=NC
10840      LINE(K)=PLUS
10850 10      IF(Y1(I).GE.YU1) GO TO 9
10860      IF(Y1(I).LT.YL1) GO TO 9
10870      K=(X-XMIN)/(XMAX-XMIN)*(NC-1)+1.5
10880      IF(K.GT.NC) K=NC
10890      LINE(K)=STAR
10900 9      CONTINUE
10910      WRITE(6,508) Z1,(LINE(K),K=1,NC),Z2
10920 7      CONTINUE
10930      WRITE(6,504)
10940      WRITE(6,3002)
10950      WRITE(6,507) XMIN,(HEAD(I),I=1,N10)
10960      RETURN
10970 504      FORMAT(1X,14(1H-),1H,,8(5H---.),1H-)
10980 507      FORMAT(3X,3HZ/L,5X,11(610.2))
10990 508      FORMAT(1X,612.3,1X,1H,41A1,1H1,F7.1)
11000 3002      FORMAT(15X,1H1,4(9X,1H1))
11010      END
11020C -----
11022      SUBROUTINE NEARB(XTEST,YTEST,ZTEST,ZM)
11024C CALCULATE THE M-TH MODAL CURRENT COEFFICIENTS FOR THE NEAR FIELD
11026C ANALYSIS (EQS 67 & 71).
11028      COMPLEX U,A3,ZM(1)
11030      COMPLEX GT(82),GZ(82),GLT(82),GLZ(82),H1T(82)
11032      COMMON /WAVE/ BK
11034      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
11036      COMMON /BOT3/ DH(82),SV(82),CV(82)
11038      COMMON /BOT5/ T(160),TP(160),TZ(160)
11040      COMMON /INT/ M,RHC2,ZP
11042      EXTERNAL FUNCT,FUNCZ,FUNCT,FUNC1Z,FUNC2T
11044      DATA PI,U /3.14159265,(0.,1.)/
11046      BKL=BK*BL
11048      KG=NP-1
11050      NM=(NP-3)/2
11052      ETA=376.707
11054      I1=0
11056C COMPUTATION OF GREEN FUNCTION KERNELS G, H0, AND H1.
11058      DO 16 J=1,KG
11060      I1=I1+1
11062      YY=YB1(J)-YTEST
11064      XX=XB1(J)-XTEST
11066      RHO2=XX*XX+YY*YY

```

```

11068      ZP=ZTEST
11070      CALL CSIMP(FUNCT,-BL,BL,0.05,10,A3,GT(I1),K,IER)
11072      GT(I1)=GT(I1)/4./PI
11074      CALL CSIMP(FUNCT,-BL,BL,0.05,10,A3,GZ(I1),K,IER)
11076      GZ(I1)=GZ(I1)/4./PI
11078      CALL CSIMP(FUNCT,-BL,BL,0.05,10,A3,G1T(I1),K,IER)
11080      G1T(I1)=G1T(I1)/4./PI
11082      CALL CSIMP(FUNCT,-BL,BL,0.05,10,A3,G1Z(I1),K,IER)
11084      G1Z(I1)=G1Z(I1)/4./PI
11086      CALL CSIMP(FUNCT,-BL,BL,0.05,10,A3,H1T(I1),K,IER)
11088      H1T(I1)=H1T(I1)/4./PI
11090      16 CONTINUE
11092C  COMPUTATION OF M-TH MODAL CURRENT COEFFICIENTS (EQS. 67 & 71).
11094      DO 3C J=1,NM
11096      JL=(J-1)*6
11098      J3=(J-1)*4
11100      J1=2*(J-1)
11102C  ELECTRIC FIELD.
11104      L1=JL+1
11106      L2=L1+NM*6
11108      L3=L1+1
11110      L4=L2+1
11112      L5=L1+2
11114      L6=L2+2
11116      ZM(L1)=0.
11118      ZM(L2)=0.
11120      ZM(L3)=0.
11122      ZM(L4)=0.
11124      ZM(L5)=0.
11126      ZM(L6)=0.
11128C  MAGNETIC FIELD.
11130      K1=L1+3
11132      K2=L2+3
11134      K3=L1+4
11136      K4=L2+4
11138      K5=L1+5
11140      K6=L2+5
11142      ZM(K1)=0.
11144      ZM(K2)=0.
11146      ZM(K3)=0.
11148      ZM(K4)=0.
11150      ZM(K5)=0.
11152      ZM(K6)=0.
11154      DO 7C JJ=1,4
11156      J2=J1+JJ
11158      J7=J3+JJ
11160      J4=J2
11162      XX=X81(J2)-XTEST
11164      YY=Y81(J2)-YTEST
11166C  ELECTRIC FIELD.
11168C  DX COMPONENT.
11170      ZM(L1)=ZM(L1)+DH(J2)*(CV(J2)+T(J7)*GT(J4)+

```

```

11172      1 XX*TP(J7)*G1T(J4)/BK/BK)
11174      ZM(L2)=ZM(L2)+DH(J2)*XX*TZ(J7)*G1T(J4)
11176C UY COMPONENT.
11178      ZM(L3)=ZM(L3)+DH(J2)*(SV(J2)*T(J7)*GT(J4)+
11180      1 YY*TP(J7)*G1T(J4)/BK/BK)
11182      ZM(L4)=ZM(L4)+DH(J2)*YY*TZ(J7)*G1T(J4)
11184C UZ COMPONENT.
11186      ZM(L5)=ZM(L5)+DH(J2)*TP(J7)*H1T(J4)
11188      ZM(L6)=ZM(L6)+DH(J2)*TZ(J7)*(GZ(J4)+
11190      1 U*PI*H1T(J4)/BK/BK)
11192C MAGNETIC FIELD.
11194C UX COMPONENT.
11196      ZM(K1)=ZM(K1)-DH(J2)*SV(J2)*T(J7)*H1T(J4)
11198      ZM(K2)=ZM(K2)+DH(J2)*YY*TZ(J7)*G1Z(J4)
11200C UY COMPONENT.
11202      ZM(K3)=ZM(K3)+DH(J2)*CV(J2)*T(J7)*H1T(J4)
11204      ZM(K4)=ZM(K4)-DH(J2)*XX*TZ(J7)*G1Z(J4)
11206C UZ COMPONENT.
11208      ZM(K5)=ZM(K5)-DH(J2)*T(J7)*(YY*CV(J2)-
11210      1 XX*SV(J2))*G1T(J4)
11212      70 CONTINUE
11214C MULTIPLY BY CONSTANTS.
11216      ZM(L1)=U*BK*ETA*ZM(L1)
11218      ZM(L2)=-M*PI*ETA/BKL*ZM(L2)
11220      ZM(L3)=U*BK*ETA*ZM(L3)
11222      ZM(L4)=-M*PI*ETA/BKL*ZM(L4)
11224      ZM(L5)=U*ETA/BK*ZM(L5)
11226      ZM(L6)=U*BK*ETA*ZM(L6)
11228      30 CONTINUE
11230      RETURN
11232      END
13580C -----
13590      SUBROUTINE NEARCI(XTEST,YTEST,ZTEST,ZM)
13600C CALCULATE THE CAP CURRENT COEFFICIENTS FOR THE NEAR FIELD ANALYSIS.
13610      COMPLEX U,A3,A4,ZM(1)
13620      COMMON /WAVE/ BK
13630      COMMON /BOT2/ NP,BL,YB(83),XB(83),YB1(82),XB1(82)
13640      COMMON /BOT3/ DH(82),SV(82),CV(82)
13650      COMMON /BOT5/ T(160),TP(160),TZ(160)
13660      COMMON /CAP1/ NC,XC,YC,ZC(2)
13670      COMMON /CAP2/ NPR,RHOC(21),RHOC1(20),DRHOC(20)
13680      COMMON /CAP3/ TCR(36),TC1(36),TPCR(36),TPCT(36)
13690      COMMON /CAP4/ RC(83),RC1(82),AC(82),CPC(82),SPC(82)
13700      DATA PI,U /3.14159265,(0.,1.)/
13710      NM=(NP-3)/2
13720      LR=(NPR-3)/2
13730      LC=NC+NM+LR
13740      BK2=BK*BK
13750      ETA=376.707
13760      J=0
13770C NOTE, EDGE COEFFICIENTS ARE NOT INCLUDED.
13780      DO 30 JC=1,NC

```

```

13790      DO 30 JP=1,NH
13800      DO 30 JR=1,LR
13810      J=J+1
13820      J1=(J-1)*6
13830      ELECTRIC FIELD.
13840      L1=J1+1
13850      L2=L1+1
13860      L3=L1+1
13870      L4=L2+1
13880      L5=L1+2
13890      L6=L3+2
13900      ZN(L1)=0.
13910      ZN(L2)=0.
13920      ZN(L3)=0.
13930      ZN(L4)=0.
13940      ZN(L5)=0.
13950      ZN(L6)=0.
13960      MAGNETIC FIELD.
13970      K1=L1+3
13980      K2=L2+3
13990      K3=L1+4
14000      K4=L2+4
14010      K5=L1+5
14020      K6=L2+5
14030      ZN(K1)=0.
14040      ZN(K2)=0.
14050      ZN(K3)=0.
14060      ZN(K4)=0.
14070      ZN(K5)=0.
14080      ZN(K6)=0.
14090      JP2=JP+1
14100      JP7=JP+1
14110      DO 70 JP=1,4
14120      JP2=JP+1
14130      JP7=JP+1
14140      JR2=JR+1
14150      JR7=JR+1
14160      DO 70 JR=1,4
14170      JR2=JR+1
14180      JR7=JR+1
14190      AJ=AC(JP2)+ABS(RHOC(JR2+1)**2-RHOC(JR2)**2)
14200      RJ=RHOC1(JR2)+RC1(JP2)
14210      XX=XC+RJ*CPIC(JP2)-XTEST
14220      YY=YC+RJ*SPIC(JP2)-YTEST
14230      ZZ=ZC(JP2)-ZTEST
14240      RR=SQRT(XX*XX+YY*YY+ZZ*ZZ)
14250      A3=CEXP(-U*BK+RR)/4./PI/RR
14260      A4=(1./RR+U*BK)/RR*A3
14270      ELECTRIC FIELD.
14280      UX COMPONENT.
14290      ZN(L1)=ZN(L1)+AJ*TCT(JR7)+(CV(JP2)+T(JP7)+A3+
14300      1 XX/BK2+TP(JP7)/RHOC1(JR2)+A4)

```

```

14310      ZM(L2)=ZM(L2)+AJ*TZ(JP7)*(CPC(JP2)*TCR(JR7)+A3+
14320      1 XX/BK2*(TPCR(JR7)/RC1(JP2)+TCR(JR7)/RJ)*A4)
14330C UY COMPONENT.
14340      ZM(L3)=ZM(L3)+AJ*TCT(JR7)*(SV(JP2)*T(JP7)+A3+
14350      1 YY/BK2*TP(JP7)/RHOC1(JR2)+A4)
14360      ZM(L4)=ZM(L4)+AJ*TZ(JP7)*(SPC(JP2)*TCR(JR7)+A3+
14370      1 YY/BK2*(TPCR(JR7)/RC1(JP2)+TCR(JR7)/RJ)*A4)
14380C UZ COMPONENT.
14390      ZM(L5)=ZM(L5)+AJ*ZZ*TP(JP7)/RHOC1(JR2)*TCT(JR7)+A4
14400      ZM(L6)=ZM(L6)+AJ*ZZ*TZ(JP7)*
14410      1 (TPCR(JR7)/RC1(JP2)+TCR(JR7)/RJ)*A4
14420C MAGNETIC FIELD.
14430C UX COMPONENT.
14440      ZM(K1)=ZM(K1)-AJ*ZZ*SV(JP2)*T(JP7)*TCT(JR7)+A4
14450      ZM(K2)=ZM(K2)-AJ*ZZ*SPC(JP2)*TZ(JP7)*TCR(JR7)+A4
14460C UY COMPONENT.
14470      ZM(K3)=ZM(K3)+AJ*ZZ*CV(JP2)*T(JP7)*TCT(JR7)+A4
14480      ZM(K4)=ZM(K4)+AJ*ZZ*CPC(JP2)*TZ(JP7)*TCR(JR7)+A4
14490C UZ COMPONENT.
14500      ZM(K5)=ZM(K5)-AJ*T(JP7)*TCT(JR7)*
14510      1 (YY*CV(JP2)-XX*SV(JP2))+A4
14520      ZM(K6)=ZM(K6)-AJ*TZ(JP7)*TCR(JR7)*
14530      1 (YY*CPC(JP2)-XX*SPC(JP2))+A4
14540      7G CONTINUE
14550C MULTIPLY BY CONSTANTS.
14560      ZM(L1)=U*BK*ETA*ZM(L1)
14570      ZM(L2)=U*BK*ETA*ZM(L2)
14580      ZM(L3)=U*BK*ETA*ZM(L3)
14590      ZM(L4)=U*BK*ETA*ZM(L4)
14600      ZM(L5)=U*ETA/BK*ZM(L5)
14610      ZM(L6)=U*ETA/BK*ZM(L6)
14620      30 CONTINUE
14630      RETURN
14640      END
-----
14650C SUBROUTINE NEARW(XTEST,YTEST,ZTEST,ZM)
14660C CALCULATE THE WIRE/JUNCTION CURRENT COEFFICIENTS FOR THE NEAR
14670C FIELD ANALYSIS.
14680C COMPLEX U,A3,A4,ZM(1)
14690      COMMON /WAVE/ BK
14700      COMMON /WIRE1/ NPW,XW(101),YW(101),ZW(101),
14710      1 XW1(100),YW1(100),ZW1(100)
14720      COMMON /WIRE2/ OHW(100),OXW(100),OYW(100),OZW(100)
14730      COMMON /WIRE3/ NW,INOW(6),RADW(5)
14740      COMMON /WIRE4/ LW,TW(196),TPW(196),INDTW(49)
14750      COMMON /JUNC1/ NJ,INDJW(10),RADJ(10),RADD(10)
14760      COMMON /JUNC2/ TJ(20),TPJ(20),INDTJ(10)
14770      COMMON /JUNC3/ XJ(10),YJ(10),ZJ(10)
14780      COMMON /JUNC4/ UXJ(10),UYJ(10),UZJ(10)
14790      COMMON /JUNC5/ UXJ1(10),UYJ1(10),UZJ1(10)
14800      COMMON /JUNC6/ UXJ2(10),UYJ2(10),UZJ2(10)
14810      DATA PI,U /3.14159265,(0.,1.)/
14820

```

```

14830      LWJ=LW+NJ
14840      BK2=BK*BK
14850      ETA=376.707
14855C  NOTE, JUNCTION COEFFICIENTS ARE NOT INCLUDED.
14860      DO 30 J=1,LW
14870      JL=(J-1)*6
14880      J3=(J-1)*4
14890      J1=INTW(J)-1
14900C  ELECTRIC FIELD.
14910      L1=JL+1
14920      L3=L1+1
14930      L5=L1+2
14940      ZH(L1)=C.
14950      ZH(L3)=0.
14960      ZH(L5)=0.
14970C  MAGNETIC FIELD.
14980      K1=L1+3
14990      K3=L1+4
15000      K5=L1+5
15010      ZH(K1)=0.
15020      ZH(K3)=0.
15030      ZH(K5)=0.
15040      DO 70 JJ=1,4
15070      J2=J1+JJ
15080      J7=J3+JJ
15090      XX=XH1(J2)-XTEST
15100      YY=YH1(J2)-YTEST
15110      ZZ=ZH1(J2)-ZTEST
15120      RR=SQRT(XX*XX+YY*YY+ZZ*ZZ)
15130      A3=CEXP(-U*BK*RR)/4./PI/RR
15140      A4=(1./RR+U*BK)/RR*A3
15150C  ELECTRIC FIELD.
15160C  UX COMPONENT.
15170      ZH(L1)=ZH(L1)+DXW(J2)*TW(J7)*A3+
15180      1 DHW(J2)*XX*TPW(J7)/BK2*A4
15190C  UY COMPONENT.
15200      ZH(L3)=ZH(L3)+DYW(J2)*TW(J7)*A3+
15210      1 DHW(J2)*YY*TPW(J7)/BK2*A4
15220C  UZ COMPONENT.
15230      ZH(L5)=ZH(L5)+DZW(J2)*TW(J7)*A3+
15240      1 DHW(J2)*ZZ*TPW(J7)/BK2*A4
15250C  MAGNETIC FIELD.
15260C  UX COMPONENT.
15270      ZH(K1)=ZH(K1)-(DYW(J2)*ZZ-
15280      1 DZW(J2)*YY)*TW(J7)*A4
15290C  UY COMPONENT.
15300      ZH(K3)=ZH(K3)-(DXW(J2)*XX-
15310      1 DXW(J2)*ZZ)*TW(J7)*A4
15320C  UZ COMPONENT.
15330      ZH(K5)=ZH(K5)-(DXW(J2)*YY-
15340      1 DYW(J2)*XX)*TW(J7)*A4
15350      70 CONTINUE

```

```

15360C MULTIPLY BY CONSTANTS.
15370  ZM(L1)=U*BK*ETA*ZM(L1)
15380  ZM(L3)=U*BK*ETA*ZM(L3)
15390  ZM(L5)=U*BK*ETA*ZM(L5)
15400  30 CONTINUE
15410  RETURN
15420  END
15430C *****
15440  SUBROUTINE CSIMP(F,A,B,DEL,IMAX,SII,S,N,IER)
15450C *****
15460C COMPLEX SIMPSON INTEGRATION ROUTINE.
15470  COMPLEX F,SII,S,SUMK
15480  SII=(.C,.0)
15490  S=(.0,.0)
15500  N=0
15510  BA=B-A
15520  IF (BA)20,19,20
15530  19 IER=1
15540  RETURN
15550  20 IF (DEL)22,22,23
15560  22 IER=2
15570  RETURN
15580  23 IF (IMAX-1)24,24,25
15590  24 IER=3
15600  RETURN
15610  25 X=BA/2.+A
15620  NHALF=.5
15630  SUMK=F(X)*BA*2./3.
15640  S=SUMK+(F(A)+F(B))*BA/6.
15650  DO 28 I=2,IMAX
15660  SII=S
15670  S=(S-SUMK/2.)/2.
15680  NHALF=NHALF*2
15690  ANHLE=NHALF
15700  FRSTX=A+(BA/ANHLE)/2.
15710  SUMK=F(FRSTX)
15720  XK=FRSTX
15730  KLAST=NHALF-1
15740  FINC=BA/ANHLE
15750  DO 26 K=1,KLAST
15760  XK=XK+FINC
15770  26 SUMK=SUMK+F(XK)
15780  SUMK=SUMK*2.+BA/(3.*ANHLE)
15790  S=S+SUMK
15800  IF (CABS(S).EQ.0.0) GO TO 29
15810  IF ((CABS(S-SII)/CABS(S))-DEL) 29,28,28
15820  28 CONTINUE
15830  IER=4
15840  WRITE(6,1) SII,S
15850C 1
15860  FORMAT(//,29H INTEGRATION DID NOT CONVERGE,//
15870  1 50H THE PREVIOUS AND FINAL VALUES OF THE INTEGRAL ARE,
2 13H RESPECTIVELY,4E12.3)

```

```

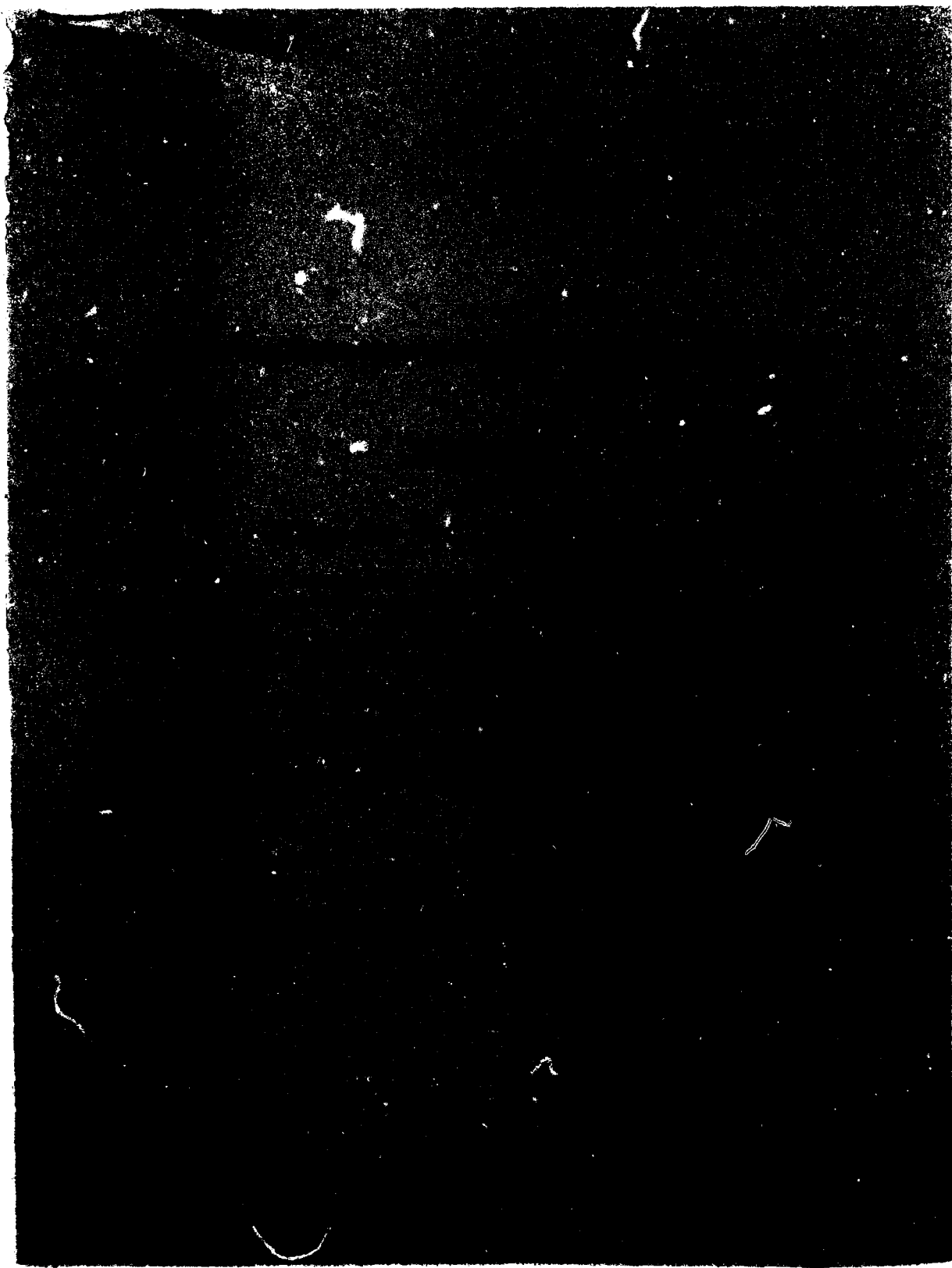
15880      GO TO 30
15890      29 IER=C
15900      30 N=2*MHALF
15910      RETURN
15920      END
15930C -----
15940      COMPLEX FUNCTION FUNCT(Z)
15950      COMPLEX U
15960      COMMON /WAVE/ BK
15970      COMMON /BOT2/ NP,BL
15980      COMMON /INT/ M,RHO2,ZP
15990      DATA PI,U /3.14159265,(0.,1.)/
16000      R=SQRT(RHO2+(Z-ZP)**2)
16010      FUNCT=CEXP(U*(M*PI*Z/BL-BK*R))/R
16020      RETURN
16030      END
16040C -----
16050      COMPLEX FUNCTION FUNCZ(Z)
16060      COMPLEX U
16070      COMMON /WAVE/ BK
16080      COMMON /BOT2/ NP,BL
16090      COMMON /INT/ M,RHO2,ZP
16100      DATA PI,U /3.14159265,(0.,1.)/
16110      R=SQRT(RHO2+(Z-ZP)**2)
16120      FUNCZ=(CEXP(U*M*PI*Z/BL)-(-1)**M)*CEXP(-U*BK*R)/R
16130      RETURN
16140      END
16150C -----
16160      COMPLEX FUNCTION FUNC1(Z)
16170      COMPLEX U
16180      COMMON /WAVE/ BK
16190      COMMON /BOT2/ NP,BL
16200      COMMON /INT/ M,RHO2,ZP
16210      DATA PI,U /3.14159265,(0.,1.)/
16220      R=SQRT(RHO2+(Z-ZP)**2)
16230      FUNC1=CEXP(U*(M*PI*Z/BL-BK*R))*((1./R+U*BK)/R/R)
16240      RETURN
16250      END
16260C -----
16270      COMPLEX FUNCTION FUNC1Z(Z)
16280      COMPLEX U
16290      COMMON /WAVE/ BK
16300      COMMON /BOT2/ NP,BL
16310      COMMON /INT/ M,RHO2,ZP
16320      DATA PI,U /3.14159265,(0.,1.)/
16330      R=SQRT(RHO2+(Z-ZP)**2)
16340      FUNC1Z=(CEXP(U*M*PI*Z/BL)-(-1)**M)*CEXP(-U*BK*R)*
16350      1 (1./R+U*BK)/R/R
16360      RETURN
16370      END
16380C -----
16390      COMPLEX FUNCTION FUNC2T(Z)

```

```

16400      COMPLEX U
16410      COMMON /WAVE/ BK
16420      COMMON /BOTZ/ MP,BL
16430      COMMON /INT/ M,RHO2,ZP
16440      DATA PI,U /3.14159265,(0.,1.)/
16450      R=SQRT(RHO2+(Z-ZP)**2)
16460      FUNC2T=CEXP(U*(M*PI*Z/BL-BK*R))*(Z-ZP)*(1./R+U*BK)/R/R
16470      RETURN
16480      END
16490/EOR

```



END

DATE
FILMED

9 82

DTIC