Studies of the factors influencing the flocculation and sedimentation of microbial cells in the treatment of Kraft mill effluents. Appendix II.

```
C***********************************************************************
C                                                                      C
C                              ********                                C
C                              * DLVO *                                C
C                              ********                                C
C                                                                      C
C                 WRITTEN BY N.DUNLOP-JONES & M.J.JAYCOCK.             C
C  DEPARTMENT OF CHEMISTRY, LOUGHBOROUGH UNIVERSITY OF TECHNOLOGY, (1981) C
C                                                                      C
C                   FOR THE PRIME INTERACTIVE SYSTEM.                  C
C                                                                      C
C    CALCULATES THE TOTAL ENERGY OF INTERACTION BETWEEN TWO NON-IDENTICAL C
C     SPHERES, WITH OR WITHOUT ADSORBED LAYERS, USING DLVO THEORY.     C
C                                                                      C
C    ORIGINAL REFERENCES:-                                             C
C         1.DERJAGUIN,B.V. & LANDAU,L., ACTA PHYSICO-CHIM(USSR),       C
C            14,633 (1941).                                            C
C         2.VERWEY,E.J.W., CHEM WEEKBL., 39,563 (1942)                 C
C         3.VERWEY,E.J.W. & OVERBEEK,J.TH.G., "THEORY AND STABILITY OF C
C            LYOPHOBIC COLLOIDS.", ELSEVIER (1948)                     C
C                                                                      C
C   THE REPULSIVE ENERGY OF INTERACTION MAY BE CALCULATED USING EITHER C
C    THE HOGG, HEALY & FUERSTENAU CONSTANT POTENTIAL OR                C
C    THE WIESE & HEALY CONSTANT CHARGE EQUATION.                       C
C                                                                      C
C   THE ATTRACTIVE ENERGY IS CALCULATED USING EITHER THE VINCENT ET AL, C
C    THE HAMAKER OR THE CLAYFIELD ET AL EQUATIONS.                     C
C                                                                      C
C   DISTANCES IN ANGSTROMS.  POTENTIALS ENTERED AND OUTPUT IN MILLIVOLTS. C
C                                                                      C
C   THE EQUATIONS USED ARE ALL IN THE C.G.S. SYSTEM OF UNITS.          C
C                                                                      C
C   PROGRAM WILL NOT FIND A SECONDARY MINIMUM DEEPER THAN -25KT.       C
C    TO CHANGE THIS VALUE ALTER THE IF STATEMENT ON LINE 1097.         C
C                                                                      C
C***********************************************************************
```

```
C***************************************************************************
C                                                                          C
C    PRIME LIBRARIES USED ARE VAPPLB & LUSUBV. THE NAGF LIBRARY ALSO USED. C
C                                                                          C
C                                                                          C
C    THE  SUBROUTINES LISTED BELOW ARE CALLED IN THE MAIN PROGRAM.         C
C                                                                          C
C    UPDATE - CONTROLS THE INPUT AND UPDATE OF DATA.                       C
C    DELSS  - DETERMINES DELTA* THE CHANGE-OVER POINT FROM HS TO HL.       C
C    FIND   - CONTROLS THE CALCULATION OF DATA - CALLS SORT & MAXMIN.      C
C    INTER  - CALCULATES THE TOTAL ENERGY OF INTERACTION WITH THE OPTION   C
C               OF USING VARIOUS FORMS OF EQUATION.                        C
C    HS     - CALCULATES THE VINCENT SHORT-RANGE GEOMETRIC FUNCTION.       C
C    HL     - CALCULATES THE VINCENT LONG-RANGE GEOMETRIC FUNCTION.        C
C    HAMFUN - CALCULATES THE  HAMAKER GEOMETRIC FUNCTION.                  C
C    CLUM   - CALCULATES THE CLAYFIELD ET AL GEOMETRIC FUNCTION.           C
C    REPUL  - CALCULATES VR AT CONSTANT POTENTIAL OR CHARGE.               C
C    SORT   - SORTS DATA INTO DECREASING ORDER & DELETES REPEATED OUTPUT.  C
C    MAXMIN - LOCATES THE PRIMARY MAXIMUM & SECONDARY MINIMUM.             C
C    STABIL - CALCULATES THE STABILITY RATIO.                              C
C    FUN1   - FUNCTION FOR STABILITY RATIO CALCULATION USED IN STABIL.     C
C    FUN2   - FUNCTION FOR STABILITY RATIO CALCULATION USED IN STABIL.     C
C    VSS    - FUNCTION USED IN STABIL.                                     C
C                                                                          C
C***************************************************************************
```

```
C***********************************************************************
C
C
C     MAIN PROGRAM SEGMENT
C
      INTEGER DIRECT
      INTEGER*4 IMIN,IMAX,IBLANK,II
      IMPLICIT DOUBLE PRECISION(A-H,L,O-Z)
      DOUBLE PRECISION EDAT$A
      DIMENSION VTKT(200),VAKT(200),VRKT(200),HX(200),UP(26),IUP(4),
     &ITITLE(28),IA(5),IFILE(10),IOUT(10),II(200),YVTSS(200),YVASS(200)
     &,NZ(10),C(10),DATE(2)
      COMMON /A/UP,IUP
      COMMON /B/VTKT,VAKT,VRKT,HX,YVTSS,YVASS
      COMMON /C/DELS,DS1S2,DP1P2,DP1S2,DP2S1
      COMMON /D/RMEAN
C
C
C     CALL STATEMENT PERMITING THE USE OF A$ LIBRARY ROUTINES.
C
$INSERT SYSCOM>A$KEYS
C
C
C     INITIALISE A FORMAT VARIABLES & ARRAYS
C
      PARAMETER(IBLANK='    ',IMAX=' MAX',IMIN=' MIN')
      DATA IFILE(1)/'**'/,IOUT(1)/'$$'/,IF/'  '/
C
C
C     SET OUTPUT LINE LENGTH TO 130 CHARACTERS
C
      CALL ATTDEV(6,7,2,65)
C
C
C
      WRITE(1,1)
    1 FORMAT(//T31,8(1H*)/T31,8H* DLVO */T31,8(1H*)///
     &'WHICH TERMINAL ARE YOU WORKING AT ?'/
     &'ENTER EITHER VDU OR PRI IF AT A PRINTING TERMINAL.'//)
      READ(1,150)IDEV
      IF(IDEV.EQ.'VD')GO TO 2
      IDEV='PR'
    2 CONTINUE
C
C
C     ARE INSTRUCTIONS FOR OPERATION REQUIRED - YES OR NO
C     IF FILES ARE REQUIRED FOR NUMBER OUTPUT OUTPUT AND FOR PLOTTING
C      - USE CORRESPONDINGLY YF AND NF .
C
      NN=3
  130 WRITE(1,140)
  140 FORMAT(//'INSTRUCTIONS? - YES OR NO - DEFAULT NO')
      READ(1,150,END=720,ERR=740)INST
  150 FORMAT(1A2)
```

```
  150 FORMAT(1A2)
      CALL CASE$A(A$FUPP,INST,2)
      IF(INST.EQ.'YE')GO TO 95
      IF((INST.EQ.'YF').OR.(INST.EQ.'NF'))GO TO 10
      INST='NO'
      GO TO 95
C
C
C    NAME & OPEN I/O DATA FILES
C
   10 WRITE(1,20)
   20 FORMAT(///'ENTER OUTPUT FILE FILENAME IN A FORM ',
     &'INCLUDING YOUR USERNUMBER'/'   -IT IS A CHEMISTRY DEPARTMENT RULE
     & THAT A FILENAME SHOULD '/'    CONTAIN A USERNUMBER. OFFENDING',
     &' FILES MAY BE DELETED.')
      NN=1
      READ(1,30,END=720,ERR=740)(IFILE(J),J=2,10)
   30 FORMAT(9A2)
C
C
C    SUBROUTINE TO CHANGE ALL L.C. TO U.C. LETTERS
C
      CALL CASE$A(A$FUPP,IFILE,20)
           DO 60 I=1,10
           IF(IFILE(I).EQ.'  ')NC=2*I-2
           IF(IFILE(I).EQ.'  ')GO TO 70
   60      CONTINUE
   70 CONTINUE
        DO 90 J=2,10
        IOUT(J)=IFILE(J)
   90   CONTINUE
C
C
C     OPEN FILES FOR DATA OUTPUT.
C
      IF(EXST$A(IFILE,NC))GO TO 76
   75 CALL OPEN$A(A$WRIT+A$SAMF,IFILE,NC,2)
      CALL OPEN$A(A$WRIT+A$SAMF,IOUT,NC,3)
      IF(IF.EQ.'AP')CALL GEND$A(2)
      IF(IF.EQ.'AP')CALL GEND$A(3)
      GO TO 79
   76 WRITE(1,77)
   77 FORMAT(//'DO YOU WISH TO DELETE THE EXISTING CONTENTS OF THE FILE
     &'/'APPEND TO THE BOTTOM OF THE EXISTING FILE'/
     &'OR CREATE A NEW FILE?')
      READ(1,150)IF
      CALL CASE$A(A$FUPP,IF,2)
      IF(IF.EQ.'CR')GO TO 10
      IF(IF.EQ.'DE')GO TO 78
      IF(IF.EQ.'AP')GO TO 75
      GO TO 76
   78 CALL DELE$A(IFILE,NC)
      CALL DELE$A(IOUT,NC)
      GO TO 75
```

```
      GO TO 75
   79 IF(IF.NE.'AP') WRITE(6,80)
   80 FORMAT(1H1)
      WRITE(1,100)IFILE,IOUT
  100 FORMAT(/'YOUR OUTPUT FILE HAS BEEN OPENED AND IS CALLED :-'//
     &20X,10A2//'YOUR FILE FOR GRAPH PLOTTING HAS BEEN OPENED AND ',
     &'IS CALLED:-'//20X,10A2//)
      NN=2
C
C
C    CONTROL FOR SPOOLED OUTPUT PRINTED PAGE SIZE
C
  110 WRITE(1,120)
  120 FORMAT('SPOOLED OUTPUT "A4" SIZE OR "LP" SIZE ?  DEFAULT "LP".')
      READ(1,150,END=720,ERR=740)LIN
      CALL CASE$A(A$FUPP,LIN,2)
      IF(LIN.EQ.'A4') GO TO 125
      LIN='LP'
  125 IF(LIN.EQ.'A4')NLINES=45
      IF(LIN.EQ.'LP')NLINES=60
C
C
C     OPEN FILE DLVODATA TO READ IN INITIAL VALUES
C
   95 CALL OPEN$A(A$READ+A$SAMF,'DLVODATA',8,1)
C
C
C
C    DATA INPUT FROM DLVODATA AND UPDATE
C
  155 READ(5,160)(IUP(N),N=1,4),(UP(N),N=5,22),UP(26)
  160 FORMAT(4A2/3(6D11.5/),D11.5)
      NI=UP(14)
      IF(NI.GT.0)READ(5,170)(NZ(J),C(J),J=1,NI)
  170 FORMAT(I2,D11.5)
      CALL CLOS$A(1)
      IF((INST.EQ.'NO').OR.(INST.EQ.'NF'))GO TO 240
      WRITE(1,180)
  180 FORMAT(///'CALCULATION OF THE INTERACTION ENERGY BETWEEN TWO',
     &' NON-IDENTICAL SPHERES,'/'WITH OR WITHOUT ADSORBED LAYERS.'//
     &'THE REPULSIVE ENERGY OF INTERACTION MAY BE CALCULATED USING EITHE
     &R'/' THE HOGG, HEALY & FUERSTENAU (CONSTANT POTENTIAL) OR'/
     &' THE WIESE AND HEALY (CONSTANT CHARGE) EQUATIONS.'//
     &'THE ATTRACTIVE TERM MAY BE CALCULATED USING THE  '/
     &' VINCENT, HAMAKER OR CLAYFIELD ET AL EQUATIONS.'//
     &'ALL DISTANCES IN ANGSTROMS.'/'ALL POTENTIALS IN MILLIVOLTS.'//)
      IF((INST.EQ.'YF').OR.(INST.EQ.'NF')) WRITE(1,182)
  182 FORMAT('DATA IS OUTPUT TO TERMINAL & FILE .'//)
      WRITE(1,184)
  184 FORMAT('THE PROMPT > INDICATES A PAUSE WHEN USING A VDU. ',
     &'TO RESTART PRESS RETURN.'/)
      WRITE(1,190)
  190 FORMAT('INITIALLY THE VALUES OF ALL PARAMETERS ARE SET TO '/
     &'THOSE READ FROM THE FILE DLVODATA.'//)
```

```
     &'THOSE READ FROM THE FILE DLVODATA.'//)
      IF(IDEV.NE.'VD') GO TO 220
      WRITE(1,200)
  200 FORMAT(1H>)
      READ(1,210)PAUSE
  210 FORMAT(1A4)
C
C
C    ENTER TITLE AND RUN CODE INFORMATION
C
  220 IF((INST.NE.'NF').AND.(INST.NE.'YF')) GO TO 295
      WRITE(1,230)
  230 FORMAT('ENTER 1] UP TO 56 CHARACTERS OF TITLE INFORMATION',
     &' & PRESS RETURN'/'      2] A RUN IDENTIFICATION CODE ( MAX. 10 ',
     &'CHARACTERS), PRESS RETURN.'/'      3] A RUN NUMBER ( <1,000/), ',
     &'PRESS RETURN.'/)
      GO TO 260
  240 NN=4
      WRITE(1,250)
  250 FORMAT(/'ENTER TITLE, RUN CODE & CODE NO.'/)
  260 READ(1,270,END=720,ERR=740)(ITITLE(J),J=1,28),(IA(K),K=1,5),NUM
  270 FORMAT(28A2/5A2/I3)
C
C
C    WRITING OF THE TITLE TO THE OUTPUT FILE
C
      IF(LIN.EQ.'A4')WRITE(6,280)(ITITLE(J),J=1,28)
  280 FORMAT(1H1,10(/),21X,28A2//)
      IF(LIN.EQ.'LP')WRITE(6,290)(ITITLE(J),J=1,28)
  290 FORMAT(1H1//21X,28A2//)
  295 IHEAD=1
  300 NSTEP=0
      ISTAB='  '
C
C    ZERO ARRAYS
C
           DO 5 J=1,200
           HX(J)=0.
           VTKT(J)=0.
           VAKT(J)=0.
           VRKT(J)=0.
           YVTSS(J)=0.
    5      YVASS(J)=0.
C
C
      UP(23)=500./(10.**8)     /* INITIAL PARTICLE SEPARATION */
      UP(24)=100./(10.**8)     /* STEP SIZE                   */
      UP(25)=0.                /* FINAL PARTICLE SEPARATION   */
C
C
C    CALL OF THE UPDATE SUBROUTINE
```

```
C     CALL OF THE UPDATE SUBROUTINE
C
       CALL UPDATE(IA,INST,IDEV,DIRECT,NZ,C)
       IF(DIRECT.EQ.'FI') GO TO 650
       IF(IUP(3).EQ.'OU') IN=15.+UP(14)+0.1
       IF(IUP(3).EQ.'WI') IN=19.+UP(14)+0.1
       IF(IHEAD.EQ.1) IN=IN+4
       IF(LIN.EQ.'A4')IN=IN+10
       IF((IUP(2).EQ.'VI').AND.(IUP(3).EQ.'WI')) IN=IN+3
       IF((IUP(2).EQ.'VI').AND.(IUP(3).EQ.'OU')) IN=IN+2
C
C
C R1=UP(5)    R2=UP(6)    D1=UP(7)    D2=UP(8)    PSI1=UP(9)    PSI2=UP(10)
C TEMP=UP(11)     EPS=UP(12)     CAPPA=UP(13)     NI=UP(14)
C A11=UP(16)    A22=UP(17)    A33=UP(18)    A44=UP(19)    A55=UP(20)
C A132=UP(21)    LAMDA=UP(22)    VSF=UP(26)
C
       H=UP(23)
       HSTEP=UP(24)
       HFINAL=UP(25)
C
C
C     CALCULATE VA/KT, VR/KT & VT/KT
C
       CALL FIND(NSTEP,NMAX,NMIN)
C
C
C     INITIAL DATA OUTPUT TO TERMINAL.
C
  310 WRITE(1,320)
  320 FORMAT(/8X,1HH,7X,5HVT/KT,7X,5HVA/KT,7X,5HVR/KT,5X,
     &12HSMOLUCHOWSKI,3X,5HFUCHS/)
       NUMLIN=1
       IV=0
         DO 360 J=1,NSTEP
         II(J)=IBLANK
         IF(J.EQ.NMAX)II(J)=IMAX
         IF(J.EQ.NMIN)II(J)=IMIN
         IF((VTKT(J).GT.300.).OR.(VTKT(J).LT.-25.)) IV=1
         IF((IV.EQ.0).AND.(VAKT(J).GE.-200.)) WRITE(1,330)
     &   II(J),HX(J),VTKT(J),VAKT(J),VRKT(J),YVTSS(J),YVASS(J)
         IF((IV.EQ.0).AND.(VAKT(J).LT.-200.)) WRITE(1,336)
     &   II(J),HX(J),VTKT(J),VAKT(J),VRKT(J),YVTSS(J)
         IF(IV.EQ.1) WRITE(1,337)
     &   II(J),HX(J),VTKT(J),VAKT(J),VRKT(J)
  330    FORMAT(1A4,8PF9.2,5(1X,0PG11.4))
  336    FORMAT(1A4,8PF9.2,4(1X,0PG11.4),2X,10(1H*))
  337    FORMAT(1A4,8PF9.2,3(1X,0PG11.4),2(2X,10(1H*)))
         IF(IDEV.NE.'VD') GO TO 360
         IF(NUMLIN.EQ.22)WRITE(1,340)
         IF(NUMLIN.EQ.22)READ(1,350)PAUSE
  340    FORMAT(1H>)
  350    FORMAT(1A4)
         IF(NUMLIN.EQ.22)NUMLIN=1
```

```
      IF(NUMLIN.EQ.22)NUMLIN=1
      NUMLIN=NUMLIN+1
  360 CONTINUE
C
C
C    EXTENSION OF H RANGE IF DESIRED
C
  370 WRITE(1,380)
  380 FORMAT(/'DO YOU WANT TO EXTEND THE RANGE OF THE DATA CALCULATED ?'
     &)
      NN=5
  390 READ(1,150,END=720,ERR=740)K
      CALL CASE$A(A$FUPP,K,2)
      IF(K.EQ.'NO') GO TO 490
      IF(K.NE.'YE') GO TO 720
  410 WRITE(1,420)
  420 FORMAT(/'ENTER NEW VALUES FOR H, HSTEP & HFINAL.',/)
      NN=6
  430 READ(1,*,END=720,ERR=740)H,HSTEP,HFINAL
      TEST=H+HFINAL
      IF(((H*H).LT.(HSTEP*HSTEP)).AND.((HFINAL*HFINAL).LT.(HSTEP*HSTEP))
     &) GO TO 700
      IF(TEST.LT.1.D-4) GO TO 680
      IF(H.GT.0)GO TO 440
      H=HFINAL
  440 IF((H*H).GT.1.D-24)H=DABS(H/(10.**8))
      IF((HFINAL*HFINAL).GT.1.D-24)HFINAL=DABS(HFINAL/(10.**8))
      IF((HSTEP*HSTEP).LT.1.D-24)GO TO 450
      HSTEP=DABS(HSTEP/(10.**8))
      NSTEP1=DABS(H-HFINAL)/HSTEP+1.1
      IF(H.GT.HFINAL)HSTEP=-HSTEP
      IF((-HSTEP).GT.(H/2.))HSTEP=HSTEP/2.
      GO TO 460
  450 NSTEP1=1
  460 NSTEP2=NSTEP1+NSTEP
      NSTEP=NSTEP+1
      HX(NSTEP)=H
      WRITE(1,320)
      NUMLIN=1
      IV=0
        DO 470 NV=NSTEP,NSTEP2
        CALL INTER(HX(NV),VAKT(NV),VRKT(NV),VTKT(NV))
        IF((VTKT(NV).GE.-25.).AND.(VTKT(NV).LE.300.))
     &  YVTSS(NV)=VSS(VTKT(NV),NV)
        IF(VAKT(NV).GE.-300.) YVASS(NV)=VSS(VAKT(NV),NV)
        II(NV)=IBLANK
        IF((VTKT(NV).GT.300.).OR.(VTKT(NV).LT.-25.)) IV=1
        IF((IV.EQ.0).AND.(VAKT(NV).GE.-200.)) WRITE(1,330)
     &  II(NV),HX(NV),VTKT(NV),VAKT(NV),VRKT(NV),YVTSS(NV),YVASS(NV)
        IF((IV.EQ.0).AND.(VAKT(NV).LT.-200.)) WRITE(1,336)
     &  II(NV),HX(NV),VTKT(NV),VAKT(NV),VRKT(NV),YVTSS(NV)
        IF(IV.EQ.1) WRITE(1,337)
     &  II(NV),HX(NV),VTKT(NV),VAKT(NV),VRKT(NV)
        IF(IDEV.NE.'VD') GO TO 465
```

```
         IF(IDEV.NE.'VD') GO TO 465
         IF(NUMLIN.EQ.22) WRITE(1,340)
         IF(NUMLIN.EQ.22) READ(1,350)
         IF(NUMLIN.EQ.22) NUMLIN=1
         NUMLIN=NUMLIN+1
  465    HX(NV+1)=HX(NV)+HSTEP
         IF(HX(NV+1).LT.1E-12)NSTEP=NV
         IF(HX(NV+1).LT.1E-12)GO TO 370
  470    CONTINUE
  480 NSTEP=NSTEP2
      GO TO 370
C
C
C    SORT AND ARRANGE EXTRA POINTS IN THE DISTANCE ARRAY
C
  490 CALL SORT(NSTEP)
      CALL MAXMIN(NSTEP,VTKT,NMAX,NMIN)
C
C
C    CALCULATION OF THE STABILITY RATIO
C    WHETHER THIS WILL BE POSSIBLE OR NOT DEPENDS ON THE SHAPE OF THE
C    POTENTIAL ENERGY CURVE . IF THE MAX. IS TOO CLOSE TO THE SURFACE
C    THE CALCULATION IS NOT PERFORMED.
C
      IF(IUP(4).EQ.'NO') GO TO 520
      IF(VTKT(NMAX).GT.300.) WRITE(1,495)
  495 FORMAT('THE MAX.VALUE OF VT/KT IS >300 AND THE STABILITY RATIO'/
     &' CANNOT THEREFORE BE CALCULATED.'/
     &'A VALUE CAN BE OBTAINED USING APPROX. EQTNS. AND (VT/KT)MAX .'//)
      IF(VTKT(NMAX).GT.300.) GO TO 520
      IF(NMAX.EQ.0) GO TO 272
      GO TO 505
C
  272 WRITE(1,271)
  271 FORMAT('THE DATA TABULATED SO FAR DO NOT SHOW A MAXIMUM.'/
     &'  DO YOU STILL WISH TO ATTEMPT TO CALCULATE THE'/
     &   STABILITY CONSTANT ?  YES OR NO ?')
      READ(1,150)ICALC
      IF(ICALC.NE.'YE') GO TO 520
C
  505 WRITE(1,510)
  510 FORMAT(//'INTEGRATIONS BEING PERFORMED.')
       CALL STABIL(W,FUCHS,RO,NMAX,ERROR1,ERROR2,NOFUN1,NOFUN2,VT)
       IF(VT.GT.0) GO TO 520
       WLOG=DLOG10(W)
       SMOL=2.*FUCHS
       SMLOG=DLOG10(SMOL)
       ROLOG=DLOG10(RO)
  520  CONTINUE
C
C
```

```
C
C    DELETE NEGATIVE POINTS CLOSER IN THAN THE PRIMARY MAXIMUM
C
      IF(NMAX.EQ.0)GO TO 550
      MSTEP=0
           DO 530 ND=1,NSTEP
           IF((HX(ND).LT.HX(NMAX)).AND.(VTKT(ND).LT.VTKT(NMIN)))
     &     GO TO 540
  530      CONTINUE
      GO TO 550
  540 MSTEP=MSTEP+1
C
  550 IF((INST.NE.'YF').AND.(INST.NE.'NF'))GO TO 625
C
C
C    OUTPUT OF CODE, RUN NUMBER AND DATA TO FILE
C
      IF((IHEAD.NE.1).AND.(LIN.EQ.'A4')) WRITE(6,800)
  800 FORMAT(1H1,10(/))
      IF((IHEAD.NE.1).AND.(LIN.EQ.'LP')) WRITE(6,80)
      CALL EDAT$A(DATE)
      WRITE(6,810)(IA(K),K=1,5),NUM,DATE(1),DATE(2)
  810 FORMAT(15X,5A2,I3,T122,A8/T122,A8)
C
C    LISTING OF THE EQUATIONS USED
C
      IF(IUP(1).EQ.'CH')WRITE(6,820)
      IF(IUP(1).EQ.'PO')WRITE(6,830)
  820 FORMAT(T13,'CALCULATION USING THE HOGG, HEALY & FUERSTENAU CONSTAN
     &T CHARGE ')
  830 FORMAT(T13,'CALCULATION USING THE HOGG, HEALY & FUERSTENAU CONSTAN
     &T POTENTIAL')
      IF(IUP(2).EQ.'HA')WRITE(6,840)
      IF(IUP(2).EQ.'VI')WRITE(6,850)
      IF(IUP(2).EQ.'CL')WRITE(6,860)
  840 FORMAT(T13,'AND THE HAMAKER ATTRACTIVE EQUATIONS.'//)
  850 FORMAT(T13,'AND THE VINCENT ATTRACTIVE EQUATIONS.'//)
  860 FORMAT(T13,'AND THE CLAYFIELD, LUMB & MACKEY ATTRACTIVE EQUATIONS.
     &'//)
C
C
C    OUTPUT OF NUMERICAL DATA TO FILE
C
      PSI1=UP(9)*299.8*1000.
      PSI2=UP(10)*299.8*1000.
      WRITE(6,870) UP(5),UP(6),PSI1,PSI2
  870 FORMAT(T13,'PARTICLE RADII:  R1=',0PF10.2,'  & R2=',0PF10.2,'  ANG
     &STROMS.'/T13,'PARTICLE POTENTIALS:  PSI1=',G10.4,'  & PSI2=',G10.4
     &,'MV')
      IF(IUP(3).EQ.'WI') WRITE(6,880)UP(7),UP(8)
  880 FORMAT(T13,'ADSORBED LAYER THICKNESSES: D1=',0PF10.2,'  & D2=',
     &0PF10.2,'  ANGSTROMS.')
       WRITE(6,890)UP(16),UP(17),UP(18),UP(21),UP(22)
  890 FORMAT(/T17,'A11',T31,'A22',T45,'A33',T58,'A132',T72,'LAMDA'/
```

```
  890 FORMAT(/T17,'A11',T31,'A22',T45,'A33',T58,'A132',T72,'LAMDA'/
     &T13,4(G10.4,4X),8PG10.4//)
      IF(IUP(3).EQ.'OU') GO TO 910
      WRITE(6,900)UP(19),UP(20)
  900 FORMAT(T17,'A44',T31,'A55'/T13,2(G10.4,4X))
  910 IF(NI.GE.2) WRITE(6,920)NI,(NZ(J),C(J),J=1,NI)
  920 FORMAT(/T16,'NI',T37,'Z',T47,'C'/T16,I2,(T36,I2,T42,G10.4))
      WRITE(6,930)UP(11),UP(12),UP(13),UP(26)
  930 FORMAT(/T13,'TEMP=',F7.2,'K',24X,'EPS=',F7.2//
     &T13,'KAPPA=',G10.4,'  /CM',10X,'VISSER FACTOR=',G10.4//)
      IF((IUP(2).EQ.'VI').AND.(IUP(3).EQ.'WI')) WRITE(6,940)DS1S2,DP1P2,
     & DP1S2,DP2S1
  940 FORMAT(T14,'DEL*S1S2',T29,'DEL*P1P2',T44,'DEL*P1S2',T59,'DEL*P2S1'
     &/T14,8PF7.2,T29,8PF7.2,T44,8PF7.2,T59,8PF7.2//)
      IF((IUP(2).EQ.'VI').AND.(IUP(3).EQ.'OU')) WRITE(6,950)DELS
  950 FORMAT(T13,'DEL*=',8PF7.2//)
C
      WRITE(6,500)
  500 FORMAT(/21X,1HH,9X,5HVT/KT,10X,5HVA/KT,10X,5HVR/KT,7X,
     &12HSMOLUCHOWSKI,5X,5HFUCHS/1H+,17X,82(1H)//)
      IN=IN+3
      I3=NSTEP-MSTEP
      JMAX=I3-(NMAX-MSTEP)
      JMIN=I3-(NMIN-MSTEP)
      WRITE(7,560)(ITITLE(M),M=1,28)
  560 FORMAT(28A2)
      WRITE(7,570)I3,JMAX,JMIN
  570 FORMAT(I3,2X,I3,2X,I3)
      IV=0
        DO 620 I2=1,NSTEP
        II(I2)=IBLANK
        IF(I2.EQ.NMAX)II(I2)=IMAX
        IF(I2.EQ.NMIN)II(I2)=IMIN
        IF((VTKT(I2).GT.300.).OR.(VTKT(I2).LT.-25.)) IV=1
        IF((IV.EQ.0).AND.(VAKT(I2).GE.-200.)) WRITE(6,580)
     &   II(I2),HX(I2),VTKT(I2),VAKT(I2),VRKT(I2),YVTSS(I2),YVASS(I2)
        IF((IV.EQ.0).AND.(VAKT(I2).LT.-200.)) WRITE(6,586)
     &   II(I2),HX(I2),VTKT(I2),VAKT(I2),VRKT(I2),YVTSS(I2)
        IF(IV.EQ.1) WRITE(6,587)
     &   II(I2),HX(I2),VTKT(I2),VAKT(I2),VRKT(I2)
  580   FORMAT(12X,1A4,8PF9.2,5(4X,0PG11.4))
  586   FORMAT(12X,1A4,8PF9.2,4(4X,0PG11.4),5X,10(1H*))
  587   FORMAT(12X,1A4,8PF9.2,3(4X,0PG11.4),2(5X,10(1H*)))
        IF(I3.LT.1) GO TO 600
        WRITE(7,590)HX(I3),VTKT(I3)
  590   FORMAT(8PF12.5,2X,E12.5)
        I3=I3-1
  600   IN=IN+1
        IF(IN.LT.NLINES)GO TO 620
        WRITE(6,80)
        IN=0
        IF(LIN.EQ.'A4')WRITE(6,610)
        IF(LIN.EQ.'A4')IN=10
  610   FORMAT(10(/))
```

```
  610    FORMAT(10(/))
         WRITE(6,500)
  620    CONTINUE
  625 IF(IUP(4).EQ.'NO') GO TO 650
      IF((VTKT(NMAX).GT.300).OR.(VT.GT.0)) GO TO 650
      WRITE(1,630)W,SMOL,RO,WLOG,SMLOG,ROLOG,ERROR1,NOFUN1,ERROR2,
     &NOFUN2
  630 FORMAT(//'STABILITY RATIOS :-'/T12,'FUCHS',T23,'SMOLUCHOWSKI',
     &T37,'REERINK & OVERBEEK'/'W',T10,G9.3,T25,G9.3,T40,G9.3/
     &'LOG(W)',T10,G9.3,T25,G9.3,T40,G9.3//
     &'ERROR IN INTEGRAL EXP(VT/KT)/(S*S)=',G9.3,'    NO. OF INTERVALS='
     &,I4/
     &'ERROR IN INTEGRAL EXP(VA/KT)/(S*S)=',G9.3,'    NO. OF INTERVALS='
     &,I4//)
      IF((INST.NE.'YF').AND.(INST.NE.'NF')) GO TO 655
      IF(IN.GT.(MLINES-10))WRITE(6,80)
      WRITE(6,640)W,SMOL,RO,WLOG,SMLOG,ROLOG,ERROR1,NOFUN1,ERROR2,
     &NOFUN2
  640 FORMAT(//T13,'STABILITY RATIOS :-'/T25,'FUCHS',T36,'SMOLUCHOWSKI',
     &T50,'REERINK & OVERBEEK'/T13,'W',T23,G9.3,T38,G9.3,T53,G9.3/
     &T13,'LOG(W)',T23,G9.3,T38,G9.3,T53,G9.3//
     &T13,'ERROR IN INTEGRAL EXP(VT/KT)/(S*S)=',G9.3,
     &'  NO. OF INTERVALS=',I4/
     &T13,'ERROR IN INTEGRAL EXP(VA/KT)/(S*S)=',G9.3,
     &'  NO. OF INTERVALS=',I4//)
  650 CONTINUE
C
C
C     FUTHER CALCULATION OR TERMINATION CONTROL SECTION
C
  655 WRITE(1,660)
  660 FORMAT(/'ENTER 0 TO FINISH.'/
     &'ENTER 1 TO REPEAT WITH NEW VALUES ONLY.'/
     &'ENTER 2 TO CHANGE TITLE AND REPEAT CALCULATION WITH NEW VALUES.')
      IF((INST.EQ.'YF').OR.(INST.EQ.'NF')) WRITE(1,665)
  665 FORMAT('ENTER 3 TO FINISH & STORE VALUES OF CONSTANTS IN ',
     &'DLVODATA .')
      NN=7
      ICALC='YE'
  670 READ(1,*,END=720,ERR=740)L
      IF(L.EQ.0)GO TO 770
      IHEAD=0
      NUM=NUM+1
      IF((L.EQ.2).AND.(INST.EQ.'YE'))GO TO 220
      IF((L.EQ.2).AND.(INST.EQ.'YF'))GO TO 220
      IF((L.EQ.2).AND.(INST.EQ.'NO')) GO TO 240
      IF((L.EQ.2).AND.(INST.EQ.'NF')) GO TO 240
      IF(L.EQ.1)GO TO 300
      GO TO 770
C
C
```

```
C
C     INPUT ERRORS SECTION
C
  680 WRITE(1,690)
  690 FORMAT('BOTH H & HFINAL =0 !!   CALCULATION NOT POSSIBLE !!')
      GO TO 720
  700 WRITE(1,710)
  710 FORMAT('HSTEP > H & HFINAL !!')
  720 WRITE(1,730)
  730 FORMAT(/'DATA INPUT INCORRECT. TYPE IN WHOLE LINE AGAIN.'/)
      GO TO 760
  740 WRITE (1,750)
  750 FORMAT(/'DATA MISREAD. TYPE IN WHOLE LINE AGAIN.'/)
  760 IF(NN.EQ.1) GO TO 10
      IF(NN.EQ.2) GO TO 110
      IF(NN.EQ.3) GO TO 130
      IF(NN.EQ.4) GO TO 240
      IF(NN.EQ.5) GO TO 390
      IF(NN.EQ.6) GO TO 430
      IF(NN.EQ.7) GO TO 670
  770 CONTINUE
C
C
C     PROGRAM TERMINATION - DATA FILE SPOOLING & DELETION INSTRUCTIONS
C
      IF((INST.NE.'YF').AND.(INST.NE.'NF')) GO TO 780
      WRITE(6,80)
      CALL CLOS$A(2)
      CALL CLOS$A(3)
      IF(L.NE.3)GO TO 780
      CALL OPEN$A(A$WRIT+A$SAMF,'DLVODATA',8,2)
      WRITE(6,160)(IUP(N),N=1,4),(UP(N),N=5,22),UP(26)
      NI=UP(14)
      IF(NI.GT.0)WRITE(6,170)(NZ(J),C(J),J=1,NI)
      CALL CLOS$A(2)
  780 IF((INST.EQ.'YF').OR.(INST.EQ.'NF')) WRITE(1,790)IFILE,IFILE,IOUT
      IF((INST.NE.'YF').AND.(INST.NE.'NF')) WRITE(1,795)
  795 FORMAT(///80(1H*)//30X,'PROGRAM RUN COMPLETE!'//80(1H*)///)
  790 FORMAT(///80(1H*)//'PROGRAM RUN COMPLETE-'//3X,'SPOOL OUTPUT',
     &' BY TYPING:-'//27X,'SPOOL ',10A2,' -LP  -FTN',//3X,'THEN ',
     &'DELETE DATA FILE BY TYPING:-'//27X,'DELETE ',10A2//25X,'& DELETE
     &',10A2//'WHEN GRAPH PLOTTING USING DLVOPLOT COMPLETE.'//80(1H*)//)
      CALL EXIT
      END
C
C*************************************************************************
```

```
C*****************************************??*************************
C
C
C     SUBROUTINE UPDATE CONTROLS THE INPUT AND UPDATE OF DATA
C
C
      SUBROUTINE UPDATE(IA,INST,IDEV,DIRECT,NZ,C)
      INTEGER DIRECT
      IMPLICIT DOUBLE PRECISION(A-H,L,O-Z)
      DOUBLE PRECISION NR1,NR2
      DIMENSION UP(26),IUP(4),NZ(10),C(10),IA(5),
     &IWORD(40),IDUM(40),IBLANK(40)
      COMMON /A/UP,IUP
      COMMON /C/DELS,DS1S2,DP1P2,DP1S2,DP2S1
$INSERT SYSCOM>A$KEYS
C
C
C     INITIALISE DATA FOR THE CALCULATION OF KAPPA
C
      PARAMETER(E=4.8029D-10,BOLTZ=1.3805D-16,AVO=6.0223D+23,PI=3.14159)
           DO 5 N=1,40
           IBLANK(N)='  '
           IWORD(N)='  '
    5      IDUM(N)='  '
      IF((UP(16).NE.0).OR.(UP(17).NE.0)) UP(21)=0.
C
C
C     CONVERSION OF UNITS
C
      UP(5)=UP(5)*(10.**8)
      UP(6)=UP(6)*(10.**8)
      UP(7)=UP(7)*(10.**8)
      UP(8)=UP(8)*(10.**8)
      UP(9)=UP(9)*299.8*1000.
      UP(10)=UP(10)*299.8*1000.
      UP(22)=UP(22)*(10.**8)
      UP(23)=UP(23)*(10.**8)
      UP(24)=UP(24)*(10.**8)
      NI=UP(14)
C
C
C     DATA INPUT
C
C
    8 MZ=1
      IF((INST.NE.'NO').AND.(INST.NE.'NF')) GO TO 12
C
C
C     LISTING OF DATA IN THE NO INSTRUCTIONS FORM
C
      WRITE(1,13)(IUP(J),J=1,4),UP(5),UP(6),UP(9),UP(10),UP(7),UP(8),
     &(UP(K),K=11,13),NI
   13 FORMAT(//'THE PRESENT VALUES OF THE DATA ARE:'//T10,4(1A2,3X)/1H#
     &/'R1=',G10.4,T21,'R2=',G10.4,T41,'PSI1=',G10.4,T61,'PSI2=',G10.4/
```

```
     &/'R1=',G10.4,T21,'R2=',G10.4,T41,'PSI1=',G10.4,T61,'PSI2=',G10.4/
     &'D1=',G10.4,T21,'D2=',G10.4/1H#/
     &'TEMP=',G10.5,T21,'EPS=',G10.4,T41,'KAPPA=',G10.4,T61,'NI=',I2/
     &1H#)
      IF(NI.LT.2)GO TO 17
      WRITE(1,14)(NZ(J),J=1,NI)
   14 FORMAT('Z',2X,10(1X,I2,8X))
      WRITE(1,15)(C(J),J=1,NI)
   15 FORMAT('C',2X,10(G10.4,1X))
      WRITE(1,16)
   16 FORMAT(1H#)
   17 WRITE(1,18)(UP(J),J=16,22),UP(26),DELS
   18 FORMAT('A11=',G10.4,T21,'A22=',G10.4,T41,'A33=',G10.4/
     &'A44=',G10.4,T21,'A55=',G10.4,T41,'A132=',G10.4/
     &'LAMDA=',G10.4,T21,'VSF=',G10.4,T41,'DELS=',G10.4,/1H#/)
      GO TO 90
C
C
C     LISTING THE DATA IN THE YES INSTRUCTIONS FORM
C
   12 IF((INST.EQ.'NO').OR.(INST.EQ.'NF')) GO TO 90
      WRITE(1,9)
   11 WRITE(1,10)(IUP(J),J=1,4)
    9 FORMAT(//'TO CHANGE CALCULATION ENTER CODE LETTERS GIVEN IN',
     &' CAPITALS'/' OF THE EQUATIONS TO BE USED.'//
     &'TO TERMINATE INPUT ENTER # ON A SEPARATE LINE.'///)
   10 FORMAT(T10,1A2,'    CONSTANT POTENTIAL/CHARGE EQUATION.'/
     &T10,1A2,'    USING HAMAKER/VINCENT/CLAYFIELD ATTRACTIVE EQUATIONS.
     &'/
     &T10,1A2,'    PARTICLES WITH/WITHOUT ADSORBED LAYERS.'/
     &T10,1A2,'    STABILITY CONSTANT BEING/NOT BEING CALCULATED.')
      GO TO 90
   20 MZ=2
      WRITE(1,25)
      WRITE(1,30)UP(5),UP(6),UP(9),UP(10),UP(7),UP(8)
   25 FORMAT(//'DETAILS OF CALCULATION PARAMETERS & THEIR CURRENT VALUES
     &.'/'TO CHANGE A PARAMETER VALUE ENTER PARAMETER NAME = NEW VALUE
     &.'/'TO TERMINATE INPUT ENTER # ON A SEPARATE LINE.'///)
   30 FORMAT(T5,'R1=',G10.4,T25,'THE RADIUS OF PARTICLE 1.'/
     &T5,'R2=',G10.4,T25,'THE RADIUS OF PARTICLE 2.'/
     &T5,'PSI1=',G10.4,T25,'THE POTENTIAL OF PARTICLE 1.'/
     &T5,'PSI2=',G10.4,T25,'THE POTENTIAL OF PARTICLE 2.'/
     &T5,'D1=',G10.4,T25,'THE THICKNESS OF THE ADSORBED LAYER ON PARTICL
     &E 1.'/T5,'D2=',G10.4,T25,'THE THICKNESS OF THE ADSORBED LAYER ON P
     &ARTICLE 2.')
      GO TO 90
   40 MZ=3
      NI=UP(14)
      WRITE(1,45)
      WRITE(1,50)(UP(J),J=11,13),NI
   45 FORMAT('CHARACTERISTICS OF THE MEDIUM:')
   50 FORMAT(T5,'TEMP=',F7.2,T25,'THE TEMPERATURE IN DEGREES KELVIN.'/
     &T5,'EPS=',G10.4,T25,'THE RELATIVE PERMITTIVITY OF THE MEDIUM.'/
     &T5,'KAPPA=',G10.4,T25,'DEBYE-HUCKEL PARAMETER.'/
```

```
     &T5,'KAPPA=',G10.4,T25,'DEBYE-HUCKEL PARAMETER.'/
     &T5,'NI=',I2,T25,'THE NUMBER OF ION TYPES.')
      IF(NI.LT.2) GO TO 90
      WRITE(1,60)
   60 FORMAT(T10,'Z',T30,'C')
           DO 62 J=1,NI
           WRITE(1,61)NZ(J),C(J)
   61      FORMAT(T9,I2,T25,G10.4)
   62      CONTINUE
      WRITE(1,65)
   65 FORMAT('WHERE Z IS THE VALENCY OF THE ION & C ITS ',
     &'CONCENTRATION.')
      GO TO 90
   70 MZ=4
      WRITE(1,75)
      WRITE(1,80)(UP(J),J=16,22),UP(26)
   75 FORMAT('HAMAKER CONSTANTS OF PARTICLES & MEDIUM:')
   80 FORMAT(T5,'A11=',G10.4,T25,'HAMAKER CONSTANT OF PARTICLE 1.'/
     &T5,'A22=',G10.4,T25,'HAMAKER CONSTANT OF PARTICLE 2.'/
     &T5,'A33=',G10.4,T25,'HAMAKER CONSTANT OF THE MEDIUM.'/
     &T5,'A44=',G10.4,T25,'HAMAKER CONSTANT OF LAYER ON PARTICLE 1.'/
     &T5,'A55=',G10.4,T25,'HAMAKER CONSTANT OF LAYER ON PARTICLE 2.'/
     &T5,'A132=',G10.4,T25,'COMPOSITE HAMAKER CONSTANT.'/
     &T5,'LAMDA=',G10.4,T25,'CHARACTERISTIC WAVELENGTH.'/
     &T5,'VSF=',G10.4,T25,'VISSER FACTOR IN ATTRACTIVE EQUATION.')
C
C
C    ENTRY POINT FOR DATA CHANGES
C
   90 READ(1,100)(IWORD(J),J=1,40)
  100 FORMAT(40A2)
      CALL CASE$A(A$FUPP,IWORD,80)
      NFF=0.
           DO 120 J=1,40
  120      IDUM(J)=IWORD(J)
C
C
C    SUBROUTINE IDENTIFYING CHARACTERS IN STRINGS
C
C    THE CHARACTER # MARKS DATA ENTRY TERMINATION
C
      CALL LSTR$A('#',1,IWORD,80,NF,NL)
      IF(NF.NE.0) GO TO 210
      IF(MZ.GT.1) GO TO 125
C
C
C    IUP( ) VALUE CHANGES
C
      CALL LSTR$A('PO',2,IWORD,80,NF,NL)
      IF(NF.NE.0) IUP(1)='PO'
      NFF=NFF+NF
      CALL LSTR$A('CH',2,IWORD,80,NF,NL)
      IF(NF.NE.0) IUP(1)='CH'
      NFF=NFF+NF
```

```
      NFF=NFF+NF
      CALL LSTR$A('HA',2,IWORD,80,NF,NL)
      IF(NF.NE.0) IUP(2)='HA'
      NFF=NFF+NF
      CALL LSTR$A('CL',2,IWORD,80,NF,NL)
      IF(NF.NE.0) IUP(2)='CL'
      NFF=NFF+NF
      CALL LSTR$A('VI',2,IWORD,80,NF,NL)
      IF(NF.NE.0) IUP(2)='VI'
      NFF=NFF+NF
      CALL LSTR$A('WI',2,IWORD,80,NF,NL)
      IF(NF.NE.0) IUP(3)='WI'
      NFF=NFF+NF
      CALL LSTR$A('OU',2,IWORD,80,NF,NL)
      IF(NF.NE.0) IUP(3)='OU'
      NFF=NFF+NF
      CALL LSTR$A('BE',2,IWORD,80,NF,NL)
      IF(NF.NE.0) IUP(4)='BE'
      NFF=NFF+NF
      CALL LSTR$A('NO',2,IWORD,80,NF,NL)
      IF(NF.NE.0) IUP(4)='NO'
      NFF=NFF+NF
      IF(NFF.NE.0) GO TO 90
      IF((INST.EQ.'NO').OR.(INST.EQ.'NF')) GO TO 125
      GO TO 200
C
C
C    UP( ) VALUE CHANGES
C
  125 CALL LSTR$A('=',1,IDUM,80,NF,NL)
      IF(NF.EQ.0) GO TO 580
C
C
C    SUBROUTINE BLANKING OUT THE NON-NUMBER PART OF THE STRING
C
      CALL MSUB$A(IBLANK,80,1,80,IDUM,NF,1,NF)
C
C
C    PRIME FORTRAN STATEMENT PUTS NUMBERS IN THE STRING INTO A VARIABLE
C
      DECODE(80,*,IDUM,ERR=580)VARI
      IF(MZ.EQ.3) GO TO 130
      IF(MZ.EQ.4) GO TO 180
      CALL LSTR$A('R1',2,IWORD,80,NF,NL)
      IF(NF.NE.0) UP(5)=VARI
      NFF=NFF+NF
      CALL LSTR$A('R2',2,IWORD,80,NF,NL)
      IF(NF.NE.0) UP(6)=VARI
      NFF=NFF+NF
      CALL LSTR$A('D1',2,IWORD,80,NF,NL)
      IF(NF.NE.0) UP(7)=VARI
      NFF=NFF+NF
      CALL LSTR$A('D2',2,IWORD,80,NF,NL)
      IF(NF.NE.0) UP(8)=VARI
```

```
      IF(NF.NE.0) UP(8)=VARI
      NFF=NFF+NF
      CALL LSTR$A('PSI1',4,IWORD,80,NF,NL)
      IF(NF.NE.0) UP(9)=VARI
      NFF=NFF+NF
      CALL LSTR$A('PSI2',4,IWORD,80,NF,NL)
      IF(NF.NE.0) UP(10)=VARI
      NFF=NFF+NF
      IF((INST.EQ.'NO').OR.(INST.EQ.'NF')) GO TO 130
      GO TO 200
  130 CALL LSTR$A('TEMP',4,IWORD,80,NF,NL)
      IF(NF.NE.0) UP(11)=VARI
      NFF=NFF+NF
      CALL LSTR$A('EPS',3,IWORD,80,NF,NL)
      IF(NF.NE.0) UP(12)=VARI
      NFF=NFF+NF
      CALL LSTR$A('KA',2,IWORD,80,NF,NL)
      IF(NF.NE.0) UP(13)=VARI
      NFF=NFF+NF
      IF((NF.EQ.0).OR.(UP(13).EQ.0.)) GO TO 160
           DO 150 M=1,10
           NZ(M)=0
  150      C(M)=0
  160 CALL LSTR$A('NI',2,IWORD,80,NF,NL)
      NFF=NFF+NF
      IF(NF.EQ.0) GO TO 180
      UP(14)=VARI
      NI=UP(14)
  169 WRITE(1,170)
  170 FORMAT(//'ENTER THE VALENCY & CONCENTRATION OF EACH ION',
     &' ON ONE LINE .'/'SEPARATE BY A COMMA.')
      READ(1,*) (NZ(J),C(J),J=1,NI)
C
C    CHECKING CONCENTRATION ENTRIES FOR KAPPA CALCULATION.
C
      SI=0.
      DO 175 K=1,NI
      FNZ=NZ(K)
  175 SI=SI+FNZ*C(K)
      IF(SI.GT.1D-10) GO TO 176
      GO TO 178
  176 WRITE(1,177) NI
  177 FORMAT('IONIC CONCENTRATIONS IN ERROR !'/'NI=',I2//)
      GO TO 169
  178 CONTINUE
C
      IF((INST.EQ.'NO').OR.(INST.EQ.'NF')) GO TO 180
      GO TO 200
  180 CALL LSTR$A('A11',3,IWORD,80,NF,NL)
      IF(NF.NE.0) UP(16)=VARI
      NFF=NFF+NF
      CALL LSTR$A('A22',3,IWORD,80,NF,NL)
      IF(NF.NE.0) UP(17)=VARI
      NFF=NFF+NF
```

```
      NFF=NFF+NF
      CALL LSTR$A('A33',3,IWORD,80,NF,NL)
      IF(NF.NE.0) UP(18)=VARI
      NFF=NFF+NF
      CALL LSTR$A('A44',3,IWORD,80,NF,NL)
      IF(NF.NE.0) UP(19)=VARI
      NFF=NFF+NF
      CALL LSTR$A('A55',3,IWORD,80,NF,NL)
      IF(NF.NE.0) UP(20)=VARI
      NFF=NFF+NF
      CALL LSTR$A('A132',4,IWORD,80,NF,NL)
      NFF=NFF+NF
      IF(NF.EQ.0) GO TO 195
      UP(21)=VARI
      IF(UP(21).EQ.0) GO TO 195
           DO 190 J=16,20
  190      UP(J)=0.
  195 CALL LSTR$A('LAMD',4,IWORD,80,NF,NL)
      IF(NF.NE.0) UP(22)=VARI
      NFF=NFF+NF
      CALL LSTR$A('VSF',3,IWORD,80,NF,NL)
      IF(NF.NE.0) UP(26)=VARI
      NFF=NFF+NF
  200 IF(NFF.EQ.0) GO TO 580
      GO TO 90
  210 IF((INST.EQ.'NO').OR.(INST.EQ.'NF')) GO TO 220
      IF(MZ.EQ.1) GO TO 20
      IF(MZ.EQ.2) GO TO 40
      IF(MZ.EQ.3) GO TO 70
C
C
C    TEST VALIDITY OF DATA
C
  220 NR1=UP(5)
      NR2=UP(6)
      IF(NR1.GT.NR2) GO TO 240
      TEST1=UP(5)*UP(6)*UP(11)*UP(12)*UP(13) /* R1*R2*TEMP*EPS*CAPPA */
      TEST2=UP(7)*UP(8)                      /* D1*D2 */
      TEST3=UP(16)*UP(17)*UP(18)*UP(22)      /* A11*A22*A33*LAMDA */
      TEST4=UP(19)*UP(20)                    /* A44*A55 */
      TEST5=UP(21)*UP(22)                    /* A132*LAMDA */
      TEST6=UP(9)+UP(10)
      IF(TEST1.LE.0.) GO TO 630
      IF(TEST2.LT.0.) GO TO 630
      IF((TEST2.LE.0.).AND.(IUP(3).EQ.'WI')) GO TO 630
      IF((TEST3.LE.0.).AND.(UP(21).LE.0.)) GO TO 630
      IF((TEST4.LE.0.).AND.(IUP(3).EQ.'WI')) GO TO 630
      IF((TEST5.LE.0.).AND.(TEST3.LE.0.)) GO TO 630
      IF(TEST6.EQ.0) WRITE(1,225)
  225 FORMAT('PSI1 & PSI2 ARE BOTH ZERO! CANNOT CALCULATE VR.')
      IF(TEST6.EQ.0) GO TO 630
      IF((INST.EQ.'NO').OR.(INST.EQ.'NF')) GO TO 260
      WRITE(1,230)
  230 FORMAT(//'DATA INPUT COMPLETE. THE VALUES ARE AS FOLLOWS:'//)
```

```
  230 FORMAT(//'DATA INPUT COMPLETE. THE VALUES ARE AS FOLLOWS:'//)
      GO TO 260
C
C
C    R1,R2 EXCHANGE SECTION IF R1>R2
C
  240 WRITE(1,250)
  250 FORMAT(//'THE EQUATIONS USED REQUIRE THAT R1 <= R2',
     &/3X,'-THE DATA WILL BE ADJUSTED ACCORDINGLY.')
      R1=UP(6)
      UP(6)=UP(5)
      UP(5)=R1
      PSI1=UP(10)
      UP(10)=UP(9)
      UP(9)=PSI1
      A11=UP(17)
      UP(17)=UP(16)
      UP(16)=A11
      A44=UP(20)
      UP(20)=UP(19)
      UP(19)=A44
      D1=UP(8)
      UP(8)=UP(7)
      UP(7)=D1
      GO TO 220
  260 IF(IUP(2).NE.'VI') GO TO 280
C
C
C    CALCULATION OF DELTA*
C
  270 R1=UP(5)
      R2=UP(6)
      LAMDA=UP(22)
      IF(IUP(3).EQ.'WI') GO TO 272
      CALL DELSS(R1,R2,LAMDA,DELS,DIRECT,0.)
      DELS=DELS/(10.**8)
      IF(DIRECT.EQ.'FI') GO TO 630
      GO TO 280
  272 HADS=UP(7)+UP(8)
      CALL DELSS(R1,R2,LAMDA,DP1P2,DIRECT,HADS)
      DP1P2=DP1P2/(10.**8)
      IF(DIRECT.EQ.'FI') GO TO 630
      RS1=R1+UP(7)
      RS2=R2+UP(8)
      CALL DELSS(RS1,RS2,LAMDA,DS1S2,DIRECT,0.)
      IF(DIRECT.EQ.'FI') GO TO 630
      DS1S2=DS1S2/(10.**8)
      CALL DELSS(R1,RS2,LAMDA,DP1S2,DIRECT,UP(7))
      IF(DIRECT.EQ.'FI') GO TO 630
      DP1S2=DP1S2/(10.**8)
      CALL DELSS(RS1,R2,LAMDA,DP2S1,DIRECT,UP(8))
      IF(DIRECT.EQ.'FI') GO TO 630
      DP2S1=DP2S1/(10.**8)
  280 IF(NZ(1).EQ.0) GO TO 290  /* IF KAPPA CALCULATION NOT REQUIRED */
```

```
 280 IF(NZ(1).EQ.0) GO TO 290   /* IF KAPPA CALCULATION NOT REQUIRED */
C
C
C    CALCULATION OF KAPPA THE DEBYE-HUCKEL PARAMETER
C
C    REF.:  PHYSIK Z. 24, 305 (1923)
C
     TEMP=UP(11)
     EPS=UP(12)
     SUM=0.
       DO 285 J=1,NI
 285   SUM=SUM+C(J)*NZ(J)*NZ(J)
     CAPPA=DSQRT(4.*PI*E*E*AVO*SUM/(EPS*BOLTZ*TEMP*1000.))
     UP(13)=CAPPA
C
 290 CONTINUE
     IF((UP(16).EQ.0.).OR.(UP(17).EQ.0.)) GO TO 295
     IF(IUP(3).EQ.'WI') GO TO 295
     A11=UP(16)
     A22=UP(17)
     A33=UP(18)
C
C
C    CALCULATION OF THE APPARENT HAMAKER CONSTANT
C
C     REF.:  HAMAKER, H.C., PHYSICA, 4, 1058(1937)
C
     A12=DSQRT(A11*A22)
     A13=DSQRT(A11*A33)
     A23=DSQRT(A22*A33)
     A132=A12+A33-A13-A23
     UP(21)=A132
C
C
C    OUTPUT OF DATA VALUES
C
 295 IF((INST.NE.'NO').AND.(INST.NE.'NF')) GO TO 296
     WRITE(1,13)(IUP(J),J=1,4),UP(5),UP(6),UP(9),UP(10),UP(7),UP(8),
    &(UP(K),K=11,14)
     IF(NI.LT.2)GO TO 297
     WRITE(1,14)(NZ(J),J=1,NI)
     WRITE(1,15)(C(J),J=1,NI)
     WRITE(1,16)
 297 WRITE(1,18)(UP(J),J=16,22),UP(26)
     GO TO 300
 296 WRITE(1,10)(IUP(J),J=1,4)
     WRITE(1,30)UP(5),UP(6),UP(9),UP(10),UP(7),UP(8)
     WRITE(1,50)(UP(J),J=11,13),NI
     IF(NI.LT.2)GO TO 317
     IF(IDEV.NE.'VD') GO TO 317
     WRITE(1,60)
     WRITE(1,61)(NZ(J),C(J),J=1,NI)
     WRITE(1,315)
 315 FORMAT(1H>)
```

```
  315 FORMAT(1H>)
      READ(1,316)PAUSE
  316 FORMAT(1A4)
  317 CONTINUE
      WRITE(1,80)(UP(J),J=16,22),UP(26)
  300 WRITE(1,310)
  310 FORMAT('ARE YOU SATISFIED THAT THE DATA ARE CORRECT?')
      READ(1,320)IDATA
  320 FORMAT(1A2)
      CALL CASE$A(A$FUPP,IDATA,2)
      IF(IDATA.EQ.'NO')GO TO 8
      IF(IDATA.EQ.'YE')GO TO 570
      WRITE(1,330)
  330 FORMAT('INCORRECT REPLY! ANSWER YES OR NO.')
      GO TO 300
C
C
C    CONVERSION OF UNITS
C
  570 UP(22)=UP(22)/(10.**8)
      UP(23)=UP(23)/(10.**8)
      UP(24)=UP(24)/(10.**8)
      UP(9)=UP(9)/(299.8*1000.)
      UP(10)=UP(10)/(299.8*1000.)
      UP(5)=UP(5)/(10.**8)
      UP(6)=UP(6)/(10.**8)
      IF(UP(7).EQ.0.) GO TO 571
      UP(7)=UP(7)/(10.**8)
  571 IF(UP(8).EQ.0.) GO TO 650
      UP(8)=UP(8)/(10.**8)
      GO TO 650
C
C
C    INPUT ERRORS SECTION
C
  580 WRITE(1,590)
  590 FORMAT(//'DATA INPUT INCORRECT.TYPE IN WHOLE LINE AGAIN.'/)
      GO TO 90
  630 WRITE(1,640)
  640 FORMAT(//'DATA INPUT INCORRECT/INCOMPLETE. RECHECK DATA!'/)
      GO TO 8
  650 CONTINUE
C
      RETURN
      END
C
C
C***********************************************************************
```

```
C***********************************************************************
C
C
C     CALCULATION OF THE CRITICAL CHANGE-OVER POINT, DELTA*, FOR
C           NON-IDENTICAL SPHERES USING VINCENTS EQUATIONS.
C
C     REF.:    VINCENT, B., JCIS, 42, 270 (1973).
C
C     ALL DISTANCES ARE IN ANGSTROM UNITS
C
C
      SUBROUTINE DELSS(R1,R2,LAMDA,DELS,DIRECT,HADS)
      IMPLICIT DOUBLE PRECISION(A-H,L,O-Z)
      INTEGER DIRECT
$INSERT SYSCOM>A$KEYS
C
C
C     ITERATE TO WITHIN 0.001 ANGSTROM OF DELTA* USING A HALF STEP
C
      HD=5.
      NCYCLE=0.
      HTEST=100.
      RD1=R1
      RD2=R2
      LAM=LAMDA
    1 GSTEP=5.
      HDINIT=HD
      HDFOLD=0.
   10 HSHORT=HS(RD1,RD2,HD,LAM,HADS)
      HLONG=HL(RD1,RD2,HD,LAM,HADS)
      NCYCLE=NCYCLE+1
      IF(NCYCLE.GT.1000.) GO TO 95
      IF((HSHORT.GT.0.).AND.(HLONG.GT.0.)) HDIFF=DABS(HSHORT-HLONG)
      IF((HSHORT.GT.0.).AND.(HLONG.LT.0.)) HDIFF=HSHORT+DABS(HLONG)
      IF((HSHORT.LT.0.).AND.(HLONG.GT.0.)) HDIFF=DABS(HSHORT)+HLONG
      IF((HSHORT.LT.0.).AND.(HLONG.LT.0.)) HDIFF=DABS(DABS(HSHORT)-
     &DABS(HLONG))
      IF(DABS(HTEST-HD).LT.0.001)GO TO 40
      IF(HDIFF.GT.HDFOLD) GSTEP=-GSTEP/2.
      HDFOLD=HDIFF
      HTEST=HD
      HD=HD+GSTEP
      GO TO 10
   40 DELS=HD
C
C
C     TEST FOR CONVERGENCE
C
      CONVER=(HDIFF/HSHORT)*100.
      IF(CONVER.GT.0.01)GO TO 50
      GO TO 100
   50 WRITE(1,60)HDINIT,DELS,HSHORT,HLONG
   60 FORMAT('ESTIMATION OF DEL* HAS FAILED CONVERGENCE TEST'/
     &'INITIAL VALUE DEL* =',G11.5,T40,'ESTIMATED VALUE OF DEL* =',
```

```
     &'INITIAL VALUE DEL* =',G11.5,T40,'ESTIMATED VALUE OF DEL* =',
     &G11.5,/,'HS=',G11.5,T40,'HL=',G11.5/
     &'ENTER:- CONTINUE IF YOU WISH TO CONTINUE',/
     &'        REPEAT   IF YOU WANT TO ENTER YOUR INITIAL GUESS FOR ',
     &'DEL*'/
     &'        FINISH   TO QUIT THE PROGRAM ALTOGETHER.'//)
      READ(1,70)DIRECT
   70 FORMAT(1A2)
      CALL CASE$A(A$FUPP,DIRECT,2)
      IF(DIRECT.EQ.'CO') GO TO 100
      IF(DIRECT.EQ.'RE') GO TO 80
      GO TO 100
   80 WRITE(1,90)
   90 FORMAT('ENTER NEW GUESS FOR DEL*.'//)
      READ(1,*,END=80,ERR=80)HD
      GO TO 1
   95 WRITE(1,96)
   96 FORMAT('DELSS ITERATION EXCEEDS 1000 !'//)
      DELS=HD
      NCYCLE=0
      GO TO 50
  100 CONTINUE
      RETURN
      END
C
C
C*****************************************************************************
```

```
C***********************************************************************
C
C
C    SUBROUTINE FIND CONTROLS THE CALCULATION OF DATA
C          CALLS SUBROUTINES SORT & MAXMIN
C
C
      SUBROUTINE FIND(NSTEP,NMAX,NMIN)
      IMPLICIT DOUBLE PRECISION (A-H,L,O-Z)
      DIMENSION UP(26),IUP(4),VTKT(200),VAKT(200),VRKT(200),HX(200),
     &YVTSS(200),YVASS(200)
      COMMON /A/UP,IUP
      COMMON /B/VTKT,VAKT,VRKT,HX,YVTSS,YVASS
      COMMON /C/DELS,DS1S2,DP1P2,DP1S2,DP2S1
      H=UP(23)
      HSTEP=UP(24)
      HFINAL=UP(25)
      NSTEP=DABS(H-HFINAL)/HSTEP+1.1
      IF((H.LT.1D-10).OR.(HFINAL.LT.1D-10)) NSTEP=NSTEP-1
      IF(H.GT.HFINAL)HSTEP=-HSTEP
      IF((-HSTEP).GT.(H/2.))HSTEP=HSTEP/2.
      HX(1)=H
C
C
C    CALCULATE FIRST SERIES OF RESULTS (500-100 ANGSTROMS IN
C    STEPS OF 100 ANGSTROMS.)
C
      ISIGN=0
      NI=1
      NF=NSTEP
      I=1
      HST=HSTEP
    5    DO 10 N=NI,NF
         IF(HX(N).GT.DABS(HST)) GO TO 6
         NSTEP=N-1
         GO TO 14
    6    CALL INTER(HX(N),VAKT(N),VRKT(N),VTKT(N))
         IF(VTKT(N).GT.0)ISIGN=1
         IF(VTKT(N).LT.-25.) GO TO 11
         IF((VTKT(N).GE.-25.).AND.(VTKT(N).LE.300.))
     &   YVTSS(N)=VSS(VTKT(N),N)
         IF(VAKT(N).GE.-200.) YVASS(N)=VSS(VAKT(N),N)
         HX(N+1)=HX(N)+HST
   10    CONTINUE
   11 IF(I.GT.1) GO TO 13
      IF((ISIGN.EQ.0).AND.(VTKT(N).LT.-25)) GO TO 13
C
C
C    EXTEND SERIES IN SMALLER INCREMENTS (HSTEP/10) UNTIL VT/KT NEGATIVE
C
   12 I=I+1
      IF(VTKT(NSTEP).LT.-25) GO TO 14
      HST=HST/10.
      NI=NSTEP+1
```

```
      NI=NSTEP+1
      HX(NI)=HX(NSTEP)+HST
      NSTEP=DABS(HX(NSTEP)/HST)-1+.1
      NF=NI+NSTEP
      GO TO 5
   13 IF(N.EQ.NF) NSTEP=NF
      IF(N.LT.NF) NSTEP=N
C
C
C     SORT AND FIND MAX. AND MIN. VALUES OF THE ARRAY VT/KT
C
   14 CALL SORT(NSTEP)
      CALL MAXMIN(NSTEP,VTKT,NMAX,NMIN)
      IF(DABS(HST).LT.1D-8) GO TO 15
      IF((NMAX.EQ.0).AND.(I.LT.2)) GO TO 12
      IF((NMAX.EQ.NSTEP).OR.(VTKT(NSTEP).GE.0.)) GO TO 12
C
C
C     LOCATE POSITION OF PRIMARY MAXIMUM TO WITIN 0.001 ANGSTROMS
C
   15 IF((VTKT(NSTEP).LT.0.).AND.(NMAX.EQ.0))GO TO 120
      IF(HX(NMAX).LT.1.1E-8) WRITE(1,16)
   16 FORMAT('PROGRAM CALCULATING AT UNREALISTIC DISTANCES < 1 ANGSTROM
     &'/' & MAY RUN INTO DIFFICULTIES.')
      HST=HX(NMAX)/20.
      NSTEP=NSTEP+1
      HX(NSTEP)=HX(NMAX)
      VTKT(NSTEP)=VTKT(NMAX)
      VAKT(NSTEP)=VAKT(NMAX)
      VRKT(NSTEP)=VRKT(NMAX)
      YVTSS(NSTEP)=YVTSS(NMAX)
      YVASS(NSTEP)=YVASS(NMAX)
   30 J=NSTEP
   35 HST=-HST/2.
      IF(DABS(HST).LT.1.D-11) GO TO 45
   40 NSTEP=NSTEP+1
      HX(NSTEP)=HX(NSTEP-1)+HST
      CALL INTER(HX(NSTEP),VAKT(NSTEP),VRKT(NSTEP),VTKT(NSTEP))
      IF((VTKT(NSTEP).GE.-25.).AND.(VTKT(NSTEP).LE.300.))
     &YVTSS(NSTEP)=VSS(VTKT(NSTEP),NSTEP)
      IF(VAKT(NSTEP).GE.-200.) YVASS(NSTEP)=VSS(VAKT(NSTEP),NSTEP)
      IF(VTKT(NSTEP).LT.VTKT(NSTEP-1)) GO TO 35
      GO TO 40
C
C
C     DELETE POINTS CALCULATED IN HUNTING FOR A PRIMARY MAXIMUM
C      AND STORE PRIMARY MAXIMUM VALUES
C
   45 HH=HX(NSTEP)
      VT=VTKT(NSTEP)
      VA=VAKT(NSTEP)
      VR=VRKT(NSTEP)
      YT=YVTSS(NSTEP)
      YA=YVASS(NSTEP)
```

```
      YA=YVASS(NSTEP)
        DO 48 K=J,NSTEP
        HX(K)=0.
        VTKT(K)=0.
        VRKT(K)=0.
        VAKT(K)=0.
        YVTSS(K)=0.
   48   YVASS(K)=0.
      NSTEP=J
      HX(NSTEP)=HH
      VTKT(NSTEP)=VT
      VAKT(NSTEP)=VA
      VRKT(NSTEP)=VR
      YVTSS(NSTEP)=YT
      YVASS(NSTEP)=YA
C
C
C    IF NO SECONDARY MIN. FOUND LOOK AT LARGER SEPARATIONS
C
   50 IF(NMIN.GT.1)GO TO 87
      HX(NSTEP+1)=HX(1)
      VTKT(NSTEP+1)=VTKT(1)
      VAKT(NSTEP+1)=VAKT(1)
      VRKT(NSTEP+1)=VRKT(1)
      YVTSS(NSTEP+1)=YVTSS(1)
      YVASS(NSTEP+1)=YVASS(1)
      NSTEP=NSTEP+1
      HST=DABS(HSTEP)
      J=NSTEP+1
   52 NSTEP=NSTEP+1
   55 HX(NSTEP)=HX(NSTEP-1)+HST
   60 CALL INTER(HX(NSTEP),VAKT(NSTEP),VRKT(NSTEP),VTKT(NSTEP))
      IF((VTKT(NSTEP).GE.-25).AND.(VTKT(NSTEP).LT.300))
     & YVTSS(NSTEP)=VSS(VTKT(NSTEP),NSTEP)
      IF(VAKT(NSTEP).GE.-200) YVASS(NSTEP)=VSS(VAKT(NSTEP),NSTEP)
      IF(VTKT(NSTEP).GT.VTKT(NSTEP-1)) GO TO 85
      IF(HX(NSTEP).GT.(1.D-04))GO TO 70
      GO TO 52
C
C
C    IF NO SECONDARY MIN. AT SEPARATIONS LESS THAN 10,000 ANGSTROMS
C     DELETE EXTRA POINTS FROM ARRAYS
C
   70     DO 80 K=J,NSTEP
          VTKT(K)=0.
          VAKT(K)=0.
          VRKT(K)=0.
          HX(K)=0.
          YVTSS(K)=0.
          YVASS(K)=0.
   80     CONTINUE
          NSTEP=J-1
          GO TO 110
C
```

```
C
C
C    SORT & FIND MAX. & MIN. VALUES OF THE ARRAY VT/KT
C
   85 CALL SORT(NSTEP)
      CALL MAXMIN(NSTEP,VTKT,NMAX,NMIN)
C
C
C    LOCATE POSITION OF SECONDARY MIN.TO WITHIN 0.001 ANGSTROMS
C
   87 J=NSTEP+1
   90 HST=DABS(HSTEP)/10.
      NO=1
   95 HST=-HST/2.
      IF(DABS(HST).LT.(1.D-11))GO TO 105
  100 NSTEP=NSTEP+1
      IF(NO.GT.1)HH=HX(NSTEP-1)
      IF(NO.EQ.1)HH=HX(NMIN)
      HX(NSTEP)=HH+HST
      NO=2
      CALL INTER(HX(NSTEP),VAKT(NSTEP),VRKT(NSTEP),VTKT(NSTEP))
      IF((VTKT(NSTEP).GE.-25).AND.(VTKT(NSTEP).LT.300))
     & YVTSS(NSTEP)=VSS(VTKT(NSTEP),NSTEP)
      IF(VAKT(NSTEP).GE.-200) YVASS(NSTEP)=VSS(VAKT(NSTEP),NSTEP)
      IF(VTKT(NSTEP).GT.VTKT(NSTEP-1))GO TO 95
      GO TO 100
C
C
C    DELETE POINTS CALCULATED HUNTING FOR A SECONDARY MINIMUM
C     AND STORE SECONDARY MINIMUM VALUES
C
  105 HH=HX(NSTEP)
      VT=VTKT(NSTEP)
      VA=VAKT(NSTEP)
      VR=VRKT(NSTEP)
      YT=YVTSS(NSTEP)
      YA=YVASS(NSTEP)
        DO 106 K=J,NSTEP
        HX(K)=0.
        VTKT(K)=0.
        VRKT(K)=0.
        VAKT(K)=0.
        YVTSS(K)=0.
  106   YVASS(K)=0.
      NSTEP=J
      HX(NSTEP)=HH
      VTKT(NSTEP)=VT
      VAKT(NSTEP)=VA
      VRKT(NSTEP)=VR
      YVTSS(NSTEP)=YT
      YVASS(NSTEP)=YA
C
```

```
C
C
C    SORT & FIND MAX. & MIN. VALUES OF THE ARRAY VT/KT
C
  110 CALL SORT(NSTEP)
      CALL MAXMIN(NSTEP,VTKT,NMAX,NMIN)
  120 CONTINUE
      RETURN
      END
C
C
C
C***********************************************************************
```

```
C*****************************************************************************
C
C
C    SUBROUTINE INTER CONTROLS THE CALCULATION OF
C        THE TOTAL ENERGY OF INTERACTION (VT)
C
C
      SUBROUTINE INTER (H,VAKT,VRKT,VTKT)
      IMPLICIT DOUBLE PRECISION (A-H,L,O-Z)
      DOUBLE PRECISION ISS
      DIMENSION UP(26),IUP(4)
      COMMON /A/UP,IUP
      COMMON /C/DELS,DS1S2,DP1P2,DP1S2,DP2S1
C
C
C    INITIALISE VARIABLES
C
      TEMP=UP(11)
      A132=UP(21)
      LAMDA=UP(22)
      R1=UP(5)
      R2=UP(6)
      A11=UP(16)
      A22=UP(17)
      A33=UP(18)
      VSF=UP(26)
      HADS=0.
C
      IF(IUP(3).EQ.'OU') GO TO 5
      A44=UP(19)
      A55=UP(20)
      D1=UP(7)
      D2=UP(8)
      RS1=R1+D1
      RS2=R2+D2
      HADS=D1+D2
C
    5 IF(IUP(2).EQ.'HA') GO TO 150
      IF(IUP(2).EQ.'CL') GO TO 170
      IF(IUP(3).EQ.'WI') GO TO 35
      IF(H-DELS) 10,10,20
   10 HFUN=HS(R1,R2,H,LAMDA,0.)
      GO TO 30
   20 HFUN=HL(R1,R2,H,LAMDA,0.)
   30 VA=-A132*VSF/12.*HFUN
      GO TO 200
   35 IF(H-DS1S2) 40,40,50
   40 HS1S2=HS(RS1,RS2,H,LAMDA,0.)
      GO TO 60
   50 HS1S2=HL(RS1,RS2,H,LAMDA,0.)
   60 IF(H-DP1P2) 70,70,80
   70 HP1P2=HS(R1,R2,H,LAMDA,HADS)
      GO TO 90
   80 HP1P2=HL(R1,R2,H,LAMDA,HADS)
```

```
   80 HP1P2=HL(R1,R2,H,LAMDA,HADS)
   90 IF(H-DP1S2) 100,100,110
  100 HP1S2=HS(R1,RS2,H,LAMDA,D1)
      GO TO 120
  110 HP1S2=HL(R1,RS2,H,LAMDA,D1)
  120 IF(H-DP2S1) 130,130,140
  130 HP2S1=HS(RS1,R2,H,LAMDA,D2)
      GO TO 190
  140 HP2S1=HL(RS1,R2,H,LAMDA,D2)
      GO TO 190
C
  150 IF(IUP(3).EQ.'WI') GO TO 160
      VA=-A132*VSF/12.*HAMFUN(R1,R2,H,0.)
      GO TO 200
  160 HS1S2=HAMFUN(RS1,RS2,H,0.)
      HP1P2=HAMFUN(R1,R2,H,HADS)
      HP1S2=HAMFUN(R1,RS2,H,D1)
      HP2S1=HAMFUN(RS1,R2,H,D2)
      GO TO 190
  170 IF(IUP(3).EQ.'WI') GO TO 180
      CALL CLUM(H,ISS,R1,R2,0.)
      VA=-A132*VSF*ISS
      GO TO 200
  180 CALL CLUM(H,HS1S2,RS1,RS2,0.)
      CALL CLUM(H,HP1P2,R1,R2,HADS)
      CALL CLUM(H,HP1S2,R1,RS2,D1)
      CALL CLUM(H,HP2S1,RS1,R2,D2)
  190 VA=-(HS1S2*(DSQRT(A44)-DSQRT(A33))*(DSQRT(A55)-DSQRT(A33))
     &    +HP1P2*(DSQRT(A11)-DSQRT(A44))*(DSQRT(A22)-DSQRT(A55))
     &    +HP1S2*(DSQRT(A11)-DSQRT(A44))*(DSQRT(A22)-DSQRT(A55))
     &    +HP2S1*(DSQRT(A22)-DSQRT(A55))*(DSQRT(A44)-DSQRT(A33)))
     &    *VSF/12.
  200 CALL REPUL(H,VR)
      VT=VA+VR
      VTKT=VT*(10.**16)/(1.3805*TEMP)
      VAKT=VA*(10.**16)/(1.3805*TEMP)
      VRKT=VR*(10.**16)/(1.3805*TEMP)
C
      RETURN
      END
C
C
C****************************************************************************
```

```
C***************************************************************************
C
C                         FUNCTIONS HS & HL INVOLVED IN THE
C
C     CALCULATION OF THE SHORT RANGE (HS) & LONG RANGE (HL) GEOMETRIC
C                   FUNCTIONS FOR NON-IDENTICAL SPHERES
C
C     REF.:
C       1.VINCENT, B., JCIS, 42, 270 (1973).
C       2.OSMOND, D.W.J., VINCENT, B., & WAITE, F.A.W., JCIS, 42, 262 (1973)
C
      FUNCTION HS(R1,R2,H,LAMDA,HADS)
      IMPLICIT DOUBLE PRECISION(A-H,L,O-Z)
      A=1.01
      B=0.14*2.*3.1416/LAMDA
      C=R1+R2+HADS+H
      X=H/(2.*R1)
      Y=R2/R1
      U=X*X+X*Y+X
      T1=A*(Y/U+Y/(U+Y)+2.*DLOG(U/(U+Y)))
      T2=8.*R1*R1*B*(2.*Y+(2.*U+Y)*DLOG(U/(U+Y)))/C
      HS=T1+T2
      RETURN
      END
C
C
      FUNCTION HL(R1,R2,H,LAMDA,HADS)
      IMPLICIT DOUBLE PRECISION(A-H,L,O-Z)
      PI=3.14159
      AP=2.45*LAMDA/(2.*PI)
      BP=2.04*(LAMDA/(2.*PI))*(LAMDA/(2.*PI))
      C=R1+R2+HADS+H
      X=H/(2.*R1)
      Y=R2/R1
      U=X*X+X*Y+X
      S1=AP*(Y*(1.+Y)*(1.+Y)/(U*U)+Y*(1.-Y)*(1.-Y)/((U+Y)*(U+Y)))
      S2=AP*(2.*(Y*Y-Y+1.)/(U+Y)-2.*(Y*Y+Y+1.)/U+4.*DLOG((U+Y)/U))
      S3=BP*(2./(U+Y)-2./U+(Y*Y+Y+1.)/(U*U)-(Y*Y-Y+1.)/((U+Y)*(U+Y)))
      S4=BP*(Y*(1.+Y)*(1.+Y)/(U*U*U)+Y*(1.-Y)*(1.-Y)/((U+Y)*(U+Y)*(U+Y))
     &)
      HL=(S1+S2)/(10.*C)+(S3-S4)/(60.*R1*R1)
      RETURN
      END
C
C
C***************************************************************************
```

```
C*****************************************************************************
C
C              SUBROUTINE HAMFUN WHICH CALCULATES THE
C     GEOMETRIC FUNCTION IN THE CALCULATION OF VA USING HAMAKER'S EQUATION
C
      FUNCTION HAMFUN(R1,R2,H,HADS)
      DOUBLE PRECISION H,HADS,R1,R2,X,Y,Z
      X=(H+HADS)/(R1+R2)
      Y=R1/R2
      Z=X**2.+X*Y+X
      HAMFUN=Y/Z+Y/(Z+Y)+2.*DLOG(Z/(Z+Y))
C
      RETURN
      END
C
C
C*****************************************************************************
```

```
C**********************************************************************
C
C
C     SUBROUTINE CLUM CALCULATES THE GEOMETRIC FUNCTION USING THE EQUATIONS
C                              OF CLAYFIELD ET AL
C
C      REF.: 1. CLAYFIELD, E.J., LUMB, E.C. AND MACKEY, P.H.,
C                 J. COLLOID INTERFACE SCI., 37, 382 (1971).
C             2. CASIMIR, H.B.G. AND POLDER, D., PHYS. REV., 73, 360 (1948)
C
C
      SUBROUTINE CLUM (H,ISS,R1,R2,HADS)
      IMPLICIT DOUBLE PRECISION (A-H,I,L,O-Z)
      DIMENSION UP(26),NIUP(4)
      COMMON /A/UP,NIUP
C
C
C     INITIALISE VARIABLES
C
      LAMDA=UP(22)
      PI=3.14159
      H=H+HADS
      RHO=3.*LAMDA/(2.*PI)
      A=1.01
      B=0.14*2.*PI/LAMDA
      C=2.45/(2.*PI/LAMDA)
      D=2.04/((2.*PI/LAMDA)*(2.*PI/LAMDA))
C
      H2R1=H+2.*R1
      H2R2=H+2.*R2
      H2R12=H2R1+R2*2.
      IF(RHO.GE.H2R12) GO TO 300
      IF((H2R12.GE.RHO).AND.(RHO.GT.H2R2)) GO TO 300
C
C
C     IS1 REGION OF INTEGRATION.
C
      IS1A=DLOG(H2R1*H2R2/(H*H2R12))+(R1+R2)*(1./H2R12-1./H)
     &     +(R1-R2)*(1./H2R2-1./H2R1)+R1*R2
     &     *(1./H**2+1./(H2R1*H2R1)+1./(H2R2*H2R2)+1./(H2R1+2.*R2*R2))
C
      IS1B=1./H-1./H2R1-1./H2R2+1./H2R12+(R1+R2)*(1./(H2R12*H2R12)
     &     -1./(H*H))+(R1-R2)*(1./(H2R2*H2R2)-1./(H2R1*H2R1))
     &     +2.*R1*R2*(1./H**3+1./H2R1**3+1./H2R2**3+1./H2R12**3)
C
      IS1=C/30.*IS1A-D/180.*IS1B
      IF((H2R1.GE.RHO).AND.(RHO.GT.H)) GO TO 100
      IF((H2R2.GE.RHO).AND.(RHO.GT.H2R1)) GO TO 200
      ISS=IS1
      GO TO 400
C
```

```
C
C
C    IS2 REGION OF INTEGRATION.
C
  100 RHO1=RHO-H
      ALPHA=A/(2.*RHO*RHO)-B/RHO-C/(3.*RHO**3)+D/(4.*RHO**4)
      BETA=-2.*A/(3.*RHO**3)+B/(RHO*RHO)+C/(2.*RHO**4)-2.*D/(5.*RHO**5)
      GAMMA=A/(4.*RHO**4)-B/(3.*RHO**3)-C/(5.*RHO**5)+D/(6.*RHO**6)
C
      IS2A=ALPHA*RHO1*RHO1*(R1-RHO1/3.)+BETA*(RHO1*RHO1*R1*(R2+H)
     &      +RHO1**3*(2.*R1-R2-H)/3.-RHO1**4/4.)
C
      IS2B=RHO1*RHO1*H*R1*H2R2+RHO1**3/3.*(4.*R1*(H+R2)-H*H2R2)
     &      +RHO1**4/2.*(R1-R2-H)-RHO1**5/5.
C
      IS2C=RHO1-2.*(R1+R2+H)*DLOG(RHO/H)+H*H2R1*(1./H-1./RHO)+2.*R2
     &      *(2.*(R1+H)*(1./H-1./RHO)-H*H2R1/2.*(1./(H*H)-1./(RHO*RHO)))
C
      IS2D=(RHO*RHO-H*H)/2.-2.*(R1+H)*RHO1+(H*H+2.*(R1+R2)*H+2.*R1*R2)
     &      *DLOG(RHO/H)+R2*(-RHO1-H*H2R2*(1./H-1./RHO))
C
      IS2E=DLOG(RHO/H)-2.*(R1+H)*(1./H-1./RHO)+H*H2R1/2.*(1./(H*H)
     &      -1./(RHO*RHO))+3.*R2
     &      *(-(1./H-1./RHO)+(R1+H)*(1./(H*H)-1./(RHO*RHO))-H*H2R1/3.
     &      *(1./H**3-1./RHO**3))
C
      IS2F=(1./H-1./RHO)-(R1+H)*(1./(H*H)-1./(RHO*RHO))+H*H2R1/3.*(1./
     &      H**3-1./RHO**3)-4.*R2
     &      *(0.5*(1./(H*H)-1./(RHO*RHO))-2.*(R1+H)/3.*(1./H**3-1./
     &      RHO**3)+H*H2R1/4.*(1./H**4-1./RHO**4))
C
      ISS=IS1+IS2A+GAMMA*IS2B+A/12.*IS2C-B/3.*IS2D-C/30.*IS2E+D/60.*IS2F
      GO TO 400
C
C
C    IS3 REGION OF INTEGRATION.
C
  200 ALPHA=A/(2.*RHO*RHO)-B/RHO-C/(3.*RHO**3)+D/(4.*RHO**4)
      BETA=-2.*A/(3.*RHO**3)+B/(RHO*RHO)+C/(2.*RHO**4)-2.*D/(5.*RHO**5)
      GAMMA=A/(4.*RHO**4)-B/(3.*RHO**3)-C/(5.*RHO**5)+D/(6.*RHO**6)
C
      IS3A=4.*R1**3/3.*ALPHA+4.*R1**3/3.*(R1+R2+H)*BETA
     &      +4.*R1**3*(((H*H)+2.*(R1+R2)*H+(R1+R2))+6./5.*R1*R1)*GAMMA
C
      IS3B=4.*R1-2.*(R1+R2+H)*DLOG(H2R1/H)+2.*R1*R2*(1./H+1./H2R1)
C
      IS3C=((H*H)+2.*H*(R1+R2)+2.*R1*R2)*DLOG(H2R1/H)-2.*R1*(R1+
     &      (2.*R2)+H)
C
      IS3D=DLOG(H2R1/H)-R1*(1./H+1./H2R1)+R2*(-1./H+1./H2R1+R1*(1./(H*H)
     &      +1./(H2R1*H2R1)))
C
      IS3E=1./H-1./H2R1-R1*(1./(H*H)+1./(H2R1*H2R1))+R1*(-1./(H*H)
     &      +1./(H2R1*H2R1)+2.*R1*(1./H**3+1./(H2R1**3)))
```

```
     &      +1./(H2R1*H2R1)+2.*R1*(1./H**3+1./(H2R1**3)))
C
      ISS=IS1+IS3A+A/12.*IS3B-B/3.*IS3C-C/30.*IS3D+D/180.*IS3E
      GO TO 400
C
C
C     IS4 INTEGRATION REGION.
C
  300 IS4A=R1+R2+H*DLOG(H*(H2R1+2.*R2)/(H2R1*H2R2))+R1*R2*(1./H+1./H2R1
     &      +1./H2R2+1./(H2R1+2.*R2))
C
      IS4B=(H*H+2.*(R1+R2)*H+2.*R1*R2)*DLOG(H2R1*H2R2/(H*(H2R1+2.*R2)))
     &      -4.*R1*R2
C
      IS4=A/6.*IS4A-B/3.*IS4B
      ISS=IS4
      IF(RHO.GE.H2R12) GO TO 400
C
C
C     IS5 INTEGRATION REGION.
C
      ALPHA=A/(2.*RHO*RHO)-B/RHO-C/(3.*RHO**3)+D/(4.*RHO**4)
      BETA=-2.*A/(3.*RHO**3)+B/(RHO*RHO)+C/(2.*RHO**4)-2.*D/(5.*RHO**5)
      GAMMA=A/(4.*RHO**4)-B/(3.*RHO**3)-C/(5.*RHO**5)+D/(6.*RHO**6)
      RHO1=RHO-H
      Q=2.*(2.*R2+R1+H)
      R=(H+R2)*(H+2.*R1+2.*R2)
C
      IS5A=4.*R1**3/3.-(RHO1-(2.*R2)**2)*(R1-(RHO1-2.*R2/3.))
C
      IS5B=4.*R1**3/3.*(R1+R2+H)+(RHO1-2.*R2)**4/4.-(RHO1-2.*R2)**3
     &      /3.*(2.*R1-R2-H)-(RHO1-2.*R2)**2*(R2+H)*R1
C
      IS5C=4.*R1**3/3.*((H*H)+2.*(R1+R2)*H+2.*R1*R2+6.*R1*R1/5.)
     &      -(RHO1-2.*R2)**2*H*R1*H2R2-(RHO1-2.*R2)**3/3.*(4.*R1*(H+R2)
     &      -H*H2R2)-(RHO1-2.*R2)**4/2.*(R1-R2-H)+(RHO1-2.*R2)**5/5.
C
      IS5D=-2.*(R1+R1+H)*DLOG(H2R12/RHO)+H2R12-RHO+(R*(1./RHO-1./H2R12))
     &      *(2.*R2*(-Q*(1./RHO-1./H2R12))+R/2.*(1./(RHO*RHO)-1./(H2R12
     &      *H2R12)))
C
      IS5E=(H2R12*H2R12-(RHO*RHO)/2.)-Q*(H2R12-RHO)+((H*H)+2.*(R1+R2)*H
     &      +2.*R1*R2)*DLOG(H2R12/RHO)+R2*(H2R12-RHO+R*(1./RHO-1./H2R12))
C
      IS5F=DLOG(H2R12/RHO)-Q*(1./RHO-1./H2R12)+R/2.*(1./(RHO*RHO)
     &    -1./(H2R12*H2R12))+3.*R2*(1./RHO-1./H2R12-Q/2.*(1./(RHO*RHO)
     &    -1./(H2R12*H2R12))+R/3.*(1./RHO**3-1./H2R12**3))
C
      IS5G=1./RHO-1./H2R12-Q/2.*(1./(RHO*RHO)-1./(H2R12*H2R12))+R/3.
     &      *(1./RHO**3-1./H2R12**3)+4.*R2*(0.5*(1./(RHO*RHO)
     &      -1./(H2R12*H2R12))-Q/3.*(1./RHO**3-1./H2R12**3)+R/4.
     &      *(1./RHO**4-1./H2R12**4))
C
      ISS=(IS4+ALPHA*IS5A+BETA*IS5B+GAMMA*IS5C+A/12.*IS5D-B/3.*IS5E-C
```

```
      ISS=(IS4+ALPHA*IS5A+BETA*IS5B+GAMMA*IS5C+A/12.*IS5D-B/3.*IS5E-C
     &    /30.*IS5F+D/60.*IS5G)
C
  400 CONTINUE
      ISS=ISS/(R1+H+R2)
      RETURN
      END
C
C
C***************************************************************************
```

```
C*************************************************************************
C
C                     SUBROUTINE REPUL
C    CALCULATES THE REPULSIVE ENERGY OF INTERACTION (VR)
C
C
      SUBROUTINE REPUL(H,VR)
      IMPLICIT DOUBLE PRECISION (A-H,L,O-Z)
      DIMENSION UP(26),IUP(4)
      COMMON /A/UP,IUP
C
C
C    INITIALISE VARIABLES.
C
      EPS=UP(12)
      PSI1=UP(9)
      PSI2=UP(10)
      R1=UP(5)
      R2=UP(6)
      IF(IUP(3).EQ.'OU') GO TO 10
      R1=R1+UP(7)
      R2=R2+UP(8)
   10 CAPPA=UP(13)
C
C
C    (1) CALCULATION OF THE REPULSIVE ENERGY OF INTERACTION (VR)
C                      AT CONSTANT POTENTIAL
C
C    REF.:  HOGG,R., HEALY,T.W. & FUERSTENAU,D.W.,
C           TRANS. FARADAY SOC., 62, 1638 (1966).
C
      FAC1=EPS*R1*R2*(PSI1*PSI1+PSI2*PSI2)/(4.*(R1+R2))
      FAC2=2.*PSI1*PSI2/(PSI1*PSI1+PSI2*PSI2)
      TR1=FAC1*FAC2*DLOG((1.+DEXP(-CAPPA*H))/(1.-DEXP(-CAPPA*H)))
      TR2=FAC1*DLOG(1.-DEXP(-2.*CAPPA*H))
      VR=TR1+TR2
C
C
C    (2) CALCULATION OF THE REPULSIVE ENERGY OF INTERACTION (VR)
C                      AT CONSTANT CHARGE
C
C    REF.:  WIESE,G.R. & HEALY,T.W., TRANS. FARADAY SOC., 66,490(1970)
C
      IF(IUP(1).EQ.'PO') GO TO 20
      CC1=EPS*R1*R2/(2.*(R1+R2))
      CC2=PSI1*PSI1+PSI2*PSI2
      CC3=DLOG(1.-DEXP(-2.*CAPPA*H))
      VR=VR-CC1*CC2*CC3
C
   20 CONTINUE
      RETURN
      END
C
```

```
C
C
C***********************************************************************
C
C
C                        SUBROUTINE SORT
C     SORTS DATA INTO DECREASING ORDER & DELETES REPEATED OUTPUT
C
C
      SUBROUTINE SORT(I)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION H(200),T(200),A(200),R(200),S(200),F(200)
      COMMON /B/T,A,R,H,S,F
C
C
C     DELETION OF REPEATED OUTPUT.
C
      DO 60 J=1,I
      IF(J.EQ.I)GO TO 60
      JK=J+1
   20       DO 30 K=JK,I
            IF(DABS(H(J)-H(K)).GT.(1.D-11))GO TO 30
            I=I-1
              DO 50 L=K,I
              H(L)=H(L+1)
              T(L)=T(L+1)
              A(L)=A(L+1)
              R(L)=R(L+1)
              S(L)=S(L+1)
              F(L)=F(L+1)
   50         CONTINUE
            H(L+1)=0.0
            T(L+1)=0.0
            A(L+1)=0.0
            R(L+1)=0.0
            S(L+1)=0.0
            F(L+1)=0.0
   30       CONTINUE
   60 CONTINUE
C
C
C     SORT OF THE OUTPUT ARRAY INTO DECREASING ORDER
C
      DO 130 M1=2,I
      IF(H(M1).LT.H(M1-1))GO TO 130
      WTEMP=H(M1)
      XTEMP=T(M1)
      YTEMP=A(M1)
      ZTEMP=R(M1)
      UTEMP=S(M1)
      VTEMP=F(M1)
      MJJ=I-1
            DO 80 M2=M1,MJJ
            H(M2)=H(M2+1)
```

```
                H(M2)=H(M2+1)
                T(M2)=T(M2+1)
                A(M2)=A(M2+1)
                R(M2)=R(M2+1)
                S(M2)=S(M2+1)
                F(M2)=F(M2+1)
   80           CONTINUE
                     DO 90 M3=1,MJJ
                     IF(WTEMP.GT.H(M3))GO TO 100
   90                CONTINUE
                     GO TO 130
  100 M4=I
  110 H(M4)=H(M4-1)
      T(M4)=T(M4-1)
      A(M4)=A(M4-1)
      R(M4)=R(M4-1)
      S(M4)=S(M4-1)
      F(M4)=F(M4-1)
      M4=M4-1
      IF(M4.EQ.M3)GO TO 120
      GO TO 110
  120 CONTINUE
      H(M3)=WTEMP
      T(M3)=XTEMP
      A(M3)=YTEMP
      R(M3)=ZTEMP
      S(M3)=UTEMP
      F(M3)=VTEMP
  130 CONTINUE
C
  150 CONTINUE
      RETURN
      END
C
C
C*****************************************************************************
```

```
C*****************************************************************************
C
C
C    SUBROUTINE MAXMIN FINDS THE INDICES CORRESPONDING TO THE
C           MAXIMUM AND MINIMUM VALUES OF THE ARRAY X
C
      SUBROUTINE MAXMIN(N,X,NMAX,NMIN)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(200)
      XMAX=X(1)
      XMIN=X(1)
      NMIN=1
      NMAX=1
       DO 10 J=2,N
       IF(XMAX.GT.X(J))GO TO 10
       NMAX=J
       XMAX=X(J)
   10  CONTINUE
        DO 20 L=2,N
        IF(L.GT.NMAX)GO TO 30
        IF(XMIN.LT.X(L))GO TO 20
        NMIN=L
        XMIN=X(L)
   20   CONTINUE
   30 CONTINUE
      IF(XMAX.GT.0.)GO TO 40
      NMAX=0
      NMIN=0
   40 IF(XMIN.GT.0)NMIN=0
C
      RETURN
      END
C
C
C*****************************************************************************
```

```
C*****************************************************************************
C
C
C    SUBROUTINE STABIL CALCULATES THE STABILITY RATIO
C       REF.:REERINK, H., OVERBEEK, J.TH.G., DISC.FARADAY SOC.,18,74 (1954).
C            CLENSHAW, A.R., CURTIS, NUM. MATH. 2, 197 (1960)
C            OLIVER, J., COMP. J. 15, 141 (1972)
C            O'HARA, H., SMITH, F.J., COMP. J. 11, 213 (1968)
C
      SUBROUTINE STABIL(W,FUCHS,GUESS,NMAX,ERROR1,ERROR2,NOFUN1,NOFUN2,
     &VT1)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      EXTERNAL FUN1,FUN2
      COMMON /A/UP,IUP/B/VTKT,VAKT,VRKT,HX,YVTSS,YVASS/D/RMEAN
      DIMENSION UP(26),IUP(4),VTKT(200),VAKT(200),VRKT(200),HX(200),
     &YVTSS(200),YVASS(200)
C
C
C    TESTS WHETHER THE STABILITY CONSTANT CALCULATION FEASIBLE
C
      H=1.D-8
    5 CALL INTER(H,VA,VR,VT1)
      IF(VT1.LT.0) GO TO 15
      IF(H.LE.1.D-9)GO TO 40
      H=H/10.
      GO TO 5
   15 DH=H
      GO TO 20
C
C
C    CALCULATES THE VALUE OF H WHEN VT/KT = -10.
C
   10      CALL INTER(H,VA,VR,VT)
   20      IF(VT.GT.-10.)DH=-DH/2.
           IF(DH.LT.1.D-10) GO TO 30
           H=H+DH
           GO TO 10
   30      CONTINUE
C
C
C    CALCULATES THE VALUES FOR THE STABILITY CONSTANT CALCULATION.
C
      R1=UP(5)
      R2=UP(6)
      IF(IUP(3).EQ.'OU') GO TO 35
      R1=R1+UP(7)
      R2=R2+UP(3)
   35 RMEAN=(R1+R2)/2.
      S1=(H+R1+R2)/RMEAN
      S2=(1.E-4+R1+R2)/RMEAN
C
```

```
C
C
C    CALCULATES AN INITIAL GUESS FOR THE VALUE OF W FROM THE
C                FORMULA OF REERINK & OVERBEEK
C
      GUESS=DEXP(VTKT(NMAX))/(2.*UP(13)*RMEAN)
      EPS1=GUESS/1000.
C
C
C    FUCHS INTEGRAL BY THE METHOD OF CLENSHAW & CURTIS (NAGF LIBRARY)
C
      NCOUNT=0
   70 CALL D01AGF(S1,S2,FUN1,50,EPS1,0.0,FUCHS,ERROR1,NOFUN1,0)
      NCOUNT=NCOUNT + 1
      IF(NCOUNT.GT.5)WRITE(1,75)
   75 FORMAT('NO. OF FUCHS INTEGRAL ITERATIONS>5 !'//)
      IF(NCOUNT.GT.5)GO TO 80
      IF(FUCHS.LT.(1000.*ERROR1))EPS1=FUCHS/5000.
      IF(FUCHS.LT.(1000.*ERROR1))GO TO 70
C
C
C    ESTIMATE OF THE ATTRACTIVE INTEGRAL BY THE TRAPEZIUM RULE
C
   80 AINTEG=0.
      S=S1
      DS=(S2-S1)/10.
           DO 60 I=1,10
           AINTEG=AINTEG+(FUN2(S)+FUN2(S+DS))*DS/2.
           S=S+DS
   60      CONTINUE
      EPS2=AINTEG/1000.
C
C
C    ATTRACTIVE INTEGRAL BY THE METHOD OF CLENSHAW & CURTIS
C
      CALL D01AGF(S1,S2,FUN2,50,EPS2,0.0,DENOM,ERROR2,NOFUN2,0)
C
C
C    STABILITY RATIO W EVALUATION
C
      W=FUCHS/DENOM
      WLOG=DLOG10(W)
   40 CONTINUE
      IF(VT1.GT.0) WRITE(1,50)H
   50 FORMAT(' VT/KT POSITIVE AT',F4.1,' ANGSTROM. STABILITY RATIO CALCU
     &LATION OMITTED.'/'VALUE CAN BE OBTAINED USING APPROX. EQTNS. AND ',
     &'(VT/KT)MAX.'//)
      RETURN
      END
C
C
C**********************************************************************
```

```
C*************************************************************************
C
C
C    FUNCTION USED IN THE SUBROUTINE STABIL
C
C
      FUNCTION VSS(V,I)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION UP(26),IUP(4),VTKT(200),VAKT(200),VRKT(200),HX(200),
     &YVTSS(200),YVASS(200)
      COMMON /A/UP,IUP
      COMMON /B/VTKT,VAKT,VRKT,HX,YVTSS,YVASS
      R1=UP(5)+UP(7)
      R2=UP(6)+UP(8)
      S=(HX(I)+R1+R2)/((R1+R2)/2.)
      VSS=DEXP(V)/(S*S)
      RETURN
      END
C
C
COPYRIGHT   03.12.1981 : NDJ & MJJ.
C*************************************************************************
```

INDEX FOR THE PROGRAM DLVO

TUE, 02
15:06:22

***