```
!   ijm=ij-2
!   if(ijm)777,771,772
! 777   par1=30.
!  771  par1=-120.
!  772   par1=0.
!   continue

    implicit real*8 (a-h,o-z)
    REAL*8 KSQ,KWN
    LOGICAL ERROR
       INTEGER IN
        CHARACTER*20 FILENAME
    COMMON/SPIN/S,LSTYPE,ISPIN
       COMMON/ OPTOUT /PARAM(13),RMAX,DR,SR(250,4),SI(250,4),NINT,MWVS
       COMMON /MXL/LMX,LMN,LSET,LDUM
       COMMON /OPTIN/ PARIN(13),PARMIN(13),PARMAX(13),AP3,
     $  ATHIRD,CMU,GAMMA,KWN,KSQ,DRIN,RMAXIN,AMP,AMT,alpar
      DATA PARMAX/1.0D3,5.0D0,3.0D0,1.0D3,1.0D3,5.0D0,3.0D0,3.0D1,3.0D1,
     $5.0D0,3.0D0,5.0D0,1.0D4/
C      DATA PNAME /2HUS,2HRS,2HAS,2HWS,2HWD,2HRI,2HAI,3HUSP,3HWSP,3HRSP,
C     $3HASP,2HRC,4HNORM/
      DATA DRIN /0.031D0/, RMAXIN /20.0D0/
      DATA LMX,LMN,LSET/3,1,1/
      DATA PARMIN /0.0D0,-1.0D1,1.0D-3,-1.0D3,-1.0D3,-1.0D1,1.0D-3,
     $ -1.5D1,-1.5D1,-1.0D1,0.0D0,-1.0D1,1.0D-5/
    read(5,*)FILENAME
    write(*,*)'fichier input'
        IN=INDEX(FILENAME,' ')

      OPEN(UNIT=10,FILE=FILENAME(1:IN-1)//'.inp',STATUS='OLD')
      OPEN(UNIT=8,FILE='smali.dat',STATUS='NEW',ACCESS='SEQUENTIAL')
c      OPEN(UNIT=9,FILE='SMAbe.Top',STATUS='NEW',ACCESS='SEQUENTIAL')
c      OPEN(UNIT=12,FILE='SMApot.DAT',STATUS='NEW',ACCESS='SEQUENTIAL')
      LSTYPE=1
      ispin=2
      S=.5
      READ(10,*)AMT,ZT,UR,RS,AS,ws,RI,AI,USP,RSP,ASP,maxe,dein,alpar
c 9 3    39.8333 1.273987 .75 .0 1.368 .3   7.07 1.273987 .75 300 .017  3.33

      AMP=1.
    zp=0.
      ATHIRD=AMT**(1.0D0/3.0D0)
      AP3=AMP**(1.0D0/3.0D0)
      CMU=.04783258D0*AMP*AMT/(AMP+AMT)
        PARIN(1)=uR
    PARIN(3)=AS
    PARIN(4)=ws
        PARIN(7)=AI
    PARIN(8)=USP
    PARIN(9)=0.!wsp
    PARIN(10)=RSP
    PARIN(11)=ASP
    PARIN(12)=1.
    PARIN(13)=1.
c        AEF=EEF
    DO 111 I=1,maxe
    EINC=REAL(i)*dein
        ij=einc-20
         parin(1)=ur!-.145*einc!-24./9.!24*(N-Z)/A Perey
    if(ij)20,30,30
   20   PARIN(2)=RS
        PARIN(6)=Ri
        go to 12
   30   PARIN(2)=RS!-0.005*ij
        PARIN(6)=Ri!-0.0025*ij
    parin(4)=0.!einc*0.375-7.5!WV
        go to 12
   12   PARIN(5)=ws!+.365*EINC !WS=13.5
    if(einc.gt.40)then
    parin(4)=0.!7.5-.02*(einc-40)
    parin(5)=ws!1.666!16.266-0.1*(einc-40)
    endif
        if(einc.gt.120)then
        parin(4)=0!5.9
        parin(5)=ws!1.666!8.226-0.07*(einc-120)
        endif
    a1=parin(1)
    a2=parin(2)
    a5=parin(5)
    a4=parin(4)
c   print *,einc,a1,a2,a5,a4
        ECM=EINC*AMT/(AMP+AMT)
        GAMMA=.157454D0*ZP*ZT*SQRT(AMP/EINC)
        KSQ=CMU*ECM
        KWN=SQRT(KSQ)
    CALL OPTMOD(.TRUE.,ERROR)
c   if(einc.eq.78)then
c       write(8,*)'ef=78MeV'
c   endif
c   write(8,*)einc
        if(i.eq.1)go to 12222
        if((si(1,1)*sdfc).lt.0)then
          write(*,*)'resonance s'
          write(*,*)einc
          endif


        if((si(2,1)*sdfa).lt.0)then
          write(*,*)'resonance p1/2'
          write(*,*)einc
          endif

        if((si(2,2)*sdfb).lt.0)then
          write(*,*)'resonance p3/2'
          write(*,*)einc
          endif
12222   continue
    write(8,*)einc
            DO 888 IJ=LMN,LMX
            ab=(ij-1+.5)/kwn

                DO 889 J=1,2
                WRITE(8,*)SR(IJ,J),SI(IJ,J)
 889            CONTINUE
 888            CONTINUE
          sdfc=si(1,1)
          sdfa=si(2,1)
          sdfb=si(2,2)
 111  CONTINUE

 400    format(5f10.5)
    STOP
    END
      SUBROUTINE COULOM(ERROR,ETA,KWN,RMAX,RD,MAXL)
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL ERROR
      REAL*8 KWN,K,K1,K2,K3,K4,M1,M2,M3,M4
C *** COULOMB WAVEFUNCTIONS CALCULATED AT R=RHO BY THE
C *** CONTINUED-FRACTION METHOD OF STEED
C *** SEE BARNETT FENG STEED AND GOLDFARB COMPUTER PHYSICS COMMUN 1974
    COMMON/SPIN/S,LSTYPE,ISPIN

      COMMON /COUOUT/ SIGMA(250),SINSGC(250),COSSGC(250),
     1  FC(251),FCP(250),GC(251),GCP(250)
      DATA MINL /0/
      DATA ACCUR /1.0D-15/, STEP /100.0D0/
C
C *********************DOUBLE PRECISION INSERTIONS**********************
```

```
C
      FLOAT(MMM)=DBLE(MMM)
      EXP(X)=DEXP(X)
      ALOG(X)=DLOG(X)
      ABS(X)=DABS(X)
      IFIX(X)=IDINT(X)
      SQRT(X)=DSQRT(X)
      SIN(X)=DSIN(X)
      COS(X)=DCOS(X)
      ATAN(X)=DATAN(X)
C

C ************************************************************************
c correction 18 july 89,ISPIN=2 ON 4 SEP 89
    ispin=2
    LSTYPE=1
C
C     PHASE SHIFTS FROM HIGH L SERIES
C
      ETASQ=ETA*ETA
      IF(MAXL-50)20,35,35
20    ELP=50.0D0
      J=50
      GO TO 45
35    ELP=FLOAT(MAXL)
      J=MAXL
45    ALPHA=ATAN(ETA/ELP)
      BETA=SQRT(ETASQ+ELP**2)
      Y=ALPHA*(ELP-0.5D0)+ETA*(ALOG(BETA)-1.0D0)
     1  -SIN(ALPHA)/(12.0D0*BETA)
     1  +SIN(3.0D0*ALPHA)/(360.0D0*BETA**3)
     1  -SIN(5.0D0*ALPHA)/(1260.0D0*BETA**5)
     1  +SIN(7.0D0*ALPHA)/(1680.0D0*BETA**7)
     1  -SIN(9.0D0*ALPHA)/(1188.0D0*BETA**9)
      M=J-1
      IF(J-MAXL)65,65,70
65    SIGMA(J)=Y
      TS=2.0D0*SIGMA(J)
      SINSGC(J)=SIN(TS)
      COSSGC(J)=COS(TS)
70    DO 100 I=1,M
      ELP=ELP-1.0D0
      J=J-1
      Y=Y-ATAN(ETA/ELP)
      IF(J-MAXL)95,95,100
95    SIGMA(J)=Y
      TS=2.0D0*SIGMA(J)
      SINSGC(J)=SIN(TS)
      COSSGC(J)=COS(TS)
100   CONTINUE
      RHOMX=KWN*RMAX
9999  RHO=RHOMX
      ERROR=.FALSE.
      PACE=STEP
      ACC=ACCUR
      IF(PACE.LT.100.0) PACE=100.0D0
      IF(ACC.LT.1.0D-15.OR.ACC.GT.1.0D-6) ACC=1.0D-6
      R=RHO
      KTR=1
      LMAX=MAXL
      LMIN1=MINL+1
      XLL1=FLOAT(MINL*LMIN1)
      ETA2=ETA*ETA
      TURN=ETA+SQRT(ETA2+XLL1)
      IF(R.LT.TURN.AND.ABS(ETA).GE.1.0D-6) KTR=-1
      KTRP=KTR
      GO TO 2
1     R=TURN
      TF=F
      TFP=FP
      LMAX=MINL
      KTRP=1
2     ETAR=ETA*R
      RHO2=R*R
      PL=FLOAT(LMAX+1)
      PMX=PL+0.5D0
C
C *** CONTINUED FRACTION FOR FP(MAXL)/F(MAXL) XL IS F XLPRIME IS FP **
C
      FP=ETA/PL+PL/R
      DK=ETAR*2.0D0
      DEL=0.0D0
      D=0.0D0
      F=1.0D0
      K=(PL*PL-PL+ETAR)*(2.0D0*PL-1.0D0)
      IF(PL*PL+PL+ETAR.NE.0.0) GO TO 3
      R=R+1.0D-6
      GO TO 2
3     H=(PL*PL+ETA2)*(1.0D0-PL*PL)*RHO2
      K=K+DK+PL*PL*6.0D0
      D=1.0D0/(D*H+K)
      DEL=DEL*(D*K-1.0D0)
      IF(PL.LT.PMX) DEL=-R*(PL*PL+ETA2)*(PL+1.0D0)*D/PL
      PL=PL+1.0D0
      FP=FP+DEL
      IF(D.LT.0.0) F=-F
      IF(PL.GT.20000.0D0) GO TO 11
      IF(ABS(DEL/FP).GE.ACC) GO TO 3
      FP=F*FP
      IF(LMAX.EQ.MINL) GO TO 5
      FC(LMAX+1)=F
      FCP(LMAX+1)=FP
C
C *** DOWNWARD RECURSION TO MINL FOR F AND FP. ARRAYS GC,GCP ARE STORAGE
C
      L=LMAX
      DO 4 LP=LMIN1,LMAX
      PL=FLOAT(L)
      GC(L+1)=ETA/PL+PL/R
      GCP(L+1)=SQRT(ETA2+PL*PL)/PL
      FC(L)=(GC(L+1)*FC(L+1)+FCP(L+1))/GCP(L+1)
      FCP(L)=GC(L+1)*FC(L)-GCP(L+1)*FC(L+1)
4     L=L-1
      F=FC(LMIN1)
      FP=FCP(LMIN1)
5     IF(KTRP.EQ.-1) GO TO 1
C
C *** REPEAT FOR R=TURN IF RHO LT TURN
C
C****NOW OBTAIN P+I.Q FOR MINL FROM CONTINUED FRACTION (32)
C *** REAL ARITHMETIC TO FACILITATE CONVERSION TO REAL*8
      P=0.0D0
      Q=R-ETA
      PL=0.0D0
      AR=-(ETA2+XLL1)
      AI=ETA
      BR=2.0D0*Q
      BI=2.0D0
      WI=2.0D0*ETA
      DR=BR/(BR*BR+BI*BI)
      DI=-BI/(BR*BR+BI*BI)
      DP=-(AR*DI+AI*DR)
      DQ=(AR*DR-AI*DI)
6     P=P+DP
      Q=Q+DQ
      PL=PL+2.0D0
      AR=AR+PL
      AI=AI+WI
      BI=BI+2.0D0
      D=AR*DR-AI*DI+BR
      DI=AI*DR+AR*DI+BI
      T=1.0D0/(D*D+DI*DI)
      DR=T*D
      DI=-T*DI
      H=BR*DR-BI*DI-1.0D0
      K=BI*DR+BR*DI
      T=DP*H-DQ*K
      DQ=DP*K+DQ*H
      DP=T
      IF(PL.GT.46000.0D0) GO TO 11
      IF(ABS(DP)+ABS(DQ).GE.(ABS(P)+ABS(Q))*ACC) GO TO 6
      P=P/R
      Q=Q/R
C
C *** SOLVE FOR FP,G,GP AND NORMALISE F AT L=MINL
C
      G=(FP-P*F)/Q
      GP=P*G-Q*F
      W=1.0D0/SQRT(FP*G-F*GP)
      G=W*G
      GP=W*GP
      IF(KTR.EQ.1) GO TO 8
      F=TF
      FT=TFP
      LMAX=MAXL
C
C *** RUNGE-KUTTA INTERATION OF G(MINL) AND GP(MINL) INWARDS FROM TURN
C ***              SEE FOX AND MAYERS 1968 PG 202
C
      IF(RHO.LT.0.2*TURN) PACE=999.0D0
      R3=1.0D0/3.0D0
      H=(RHO-TURN)/(PACE+1.0D0)
      H2=0.5D0*H
      I2=IFIX(PACE+0.001D0)
      ETAH=ETA*H
      H2LL=H2*XLL1
      S=(ETAH+H2LL/R)/R-H2
7     RH2=R+H2
      T=(ETAH+H2LL/RH2)/RH2-H2
      K1=H2*GP
      M1=S*G
      K2=H2*(GP+M1)
      M2=T*(G+K1)
      K3=H*(GP+M2)
      M3=T*(G+K2)
      M3=M3+M3
      K4=H2*(GP+M3)
      RH=R+H
      S=(ETAH+H2LL/RH)/RH-H2
      M4=S*(G+K3)
      G=G+(K1+K2+K2+K3+K4)*R3
      GP=GP+(M1+M2+M2+M3+M4)*R3
      R=RH
      I2=I2-1
      IF(ABS(GP).GT.1.0D70) GO TO 11
      IF(I2.GE.0) GO TO 7
      W=1.0D0/(FP*G-F*GP)
C
C *** UPWARD RECURSION FROM GC(MINL) AND GCP(MINL),STORED VALUES ARE R,S
C *** RENORMALISE FC,FCP FOR EACH L-VALUE
C
8     GC(LMIN1)=G
      GCP(LMIN1)=GP
      IF(LMAX.EQ.MINL) GO TO 10
      DO 9 L=LMIN1,LMAX
      T=GC(L+1)
      GC(L+1)=(GC(L)*GC(L+1)-GCP(L))/GCP(L+1)
      GCP(L+1)=GC(L)*GCP(L+1)-GC(L+1)*T
      FC(L+1)=W*FC(L+1)
9     FCP(L+1)=W*FCP(L+1)
      FC(LMIN1)=FC(LMIN1)*W
      FCP(LMIN1)=FCP(LMIN1)*W
      DO 12 L=LMIN1,LMAX
      GCP(L)=KWN*GCP(L)
      FCP(L)=KWN*FCP(L)
12    CONTINUE
      GCP(LMAX+1)=KWN*GCP(LMAX+1)
      FCP(LMAX+1)=KWN*FCP(LMAX+1)
      RETURN
10    FC(LMIN1)=W*F
      FCP(LMIN1)=KWN*(W*FP)
      GCP(LMIN1)=KWN*GCP(LMIN1)
      RETURN
11    W=0.0D0
      G=0.0D0
      GP=0.0D0
      ERROR=.TRUE.
      WRITE(6,777)
777   FORMAT(/1X,'PROBLEM IN COULOMB WAVEFUNCTION CALCULATION',/)
      GO TO 8
      END

    SUBROUTINE OPTMOD (NEWINT,ERROR)
    IMPLICIT REAL*8 (A-H,O-Z)
C
C  OPTICAL MODEL CALCULATION
C         INPUT PARAMETERS ARE NEWINT,ERROR,/OPTIN/
C         NEWINT .TRUE. FOR FIRST PASS ON SEARCH OR FOR NOSEARCH
C         ERROR IS RETURNED .TRUE. IN CASE OF TROUBLE
C
    COMMON/SPIN/S,LSTYPE,ISPIN

      COMMON /OPTIN/ PARIN(13),PARMIN(13),PARMAX(13),AP3,
     $  ATHIRD,CMU,GAMMA,KWN,KSQ,DRIN,RMAXIN,AMP,AMT,alpar
       COMMON/ OPTOUT /PARAM(13),RMAX,DR,SR(250,4),SI(250,4),NINT,MWVS
      COMMON /COUOUT/ SIGMA(250),SINSGC(250),COSSGC(250),
     $  F(251),FP(250),G(251),GP(250)
       COMMON /POTS/ DRSQ,UDRSQ(1000),WDRSQ(1000),HRDRSQ(1000),
     $  HIDRSQ(1000),VCDRSQ(1000),RONE,RTWO,WSTWO,ICO
      COMMON /MICSO/ H(1000)

       COMMON /MXL/LMX,LMN,LSET,LDUM
    INTEGER CONSTR(13),WSTWO
      DIMENSION UEXP(1000),WEXP(1000)
      DIMENSION UR(7),UI(7)
C$=======================================
      DIMENSION WEXP1(1000)
      DIMENSION UEXP1(1000)
C$=======================================
      REAL*8 KWN,KSQ,NORM

      LOGICAL ERROR,NEWINT,SPNORB
      EQUIVALENCE (PARAM( 1),USX ),(PARAM( 2),RSX ),
     $             (PARAM( 3),AS  ),(PARAM( 4),WSX ),
     $             (PARAM( 5),WDX ),(PARAM( 6),RIX ),
     $             (PARAM( 7),AI  ),(PARAM( 8),USPX),
     $             (PARAM( 9),WSPX),(PARAM(10),RSPX),
     $             (PARAM(11),ASP ),(PARAM(12),RCX ),
     $             (PARAM(13),NORM)
    data constr/1,2,3,4,5,6,7,8,9,10,11,12,13/
      DATA MINT /1000/
C
C******************* HI-OPTIM ************ NOTES *********************
C
C     TOGETHER WITH THE CHANGES LISTED IN SUBROUTINE 'MAIN' THE FOLLOWING
C     CARDS CONVERT THE PROGRAM TO DOUBLE PRECISION FOR IBM 360/195 USE.
```

```
C
      EXP(X)=DEXP(X)
      ABS(X)=DABS(X)
      ALOG(X)=DLOG(X)
      IFIX(X)=IDINT(X)
      FLOAT(MMM)=DBLE(MMM)
      SQRT(X)=DSQRT(X)
      SIN(X)=DSIN(X)
      COS(X)=DCOS(X)
      ATAN(X)=DATAN(X)
C***************************************************************************
C
C
C     STATEMENT FUNCTION RMAX
C
      RMX(R,A,V,RMX0)=R+A*DLOG(DMAX1(V,1.0D-3)/(1.0D-3*ECM)*
     $ DMAX1(A,1.0D-3)*(KWN+(GAMMA+KWN*R)/RMX0))
    ispin=2
    LSTYPE=1
c correction 18 july 89,ISPIN=2 ON 4 SEP 89
      ERROR=.FALSE.
      ETA1=1.0D-8
      R12=1.0D0/12.0D0
      COR=1.0D0/30.0D0
C
C     CONSTRAINTS
C
!   print *,par1
      N=1
123   DO 10 I=N,13
      IF(MICI.NE.0.AND.I.GT.4.AND.I.LE.7) GO TO 10
       IF(MICR.NE.0.AND.I.GT.1.AND.I.LE.3) GO TO 10
       IF(MICS.NE.0.AND.I.GT.9.AND.I.LE.11) GO TO 10
      II=CONSTR(I)
      PARAM(I)=PARIN(II)
      IF (PARAM(I).GE.PARMIN(I).AND.PARAM(I).LE.PARMAX(I)) GO TO 10
      ERROR=.TRUE.
C        WRITE (6,1000) PARAM(I),PARMIN(I),PARMAX(I)
C 1000 FORMAT (35H0CALCULATION TERMINATED--PARAMETER A4,3H (=1PE9.2,18H)
C     $IS OUT OF LIMITS 3X,4H.GE.,E9.2,3X,4H.LE. E9.2 )
   10 CONTINUE
      IF(MICI.EQ.1) GO TO 12
      RIMI=ABS(RIX)*ATHIRD
      IF(RIX.LT.0.0D0) RIMI=RIMI+ABS(RIX)*AP3
      IF (WSX*RIMI /3.0D0+2.0D0*WDX*AI.GE.0.0D0) GO TO 12
        WRITE (6,1004)
 1004 FORMAT (72H0CALCULATION TERMINATED--VOLUME INTEGRAL OF IMAGINARY P
     $OTENTIAL POSITIVE)
      ERROR=.TRUE.
   12 IF (ERROR) RETURN
C
C     SPNORB
C
      IF (.NOT.NEWINT) GO TO 15
      SPNORB=.FALSE.
      MICS1=MICS+1
      GO TO (13,14),MICS1
   13 IF(ABS(RSPX).LE.0.001D0.OR.ASP.LE.0.001D0) GO TO 16
      IF(ISPIN.GT.1) SPNORB=.TRUE.
      PARMIN(11)=.001D0
      GO TO 15
   14 IF(USPX.EQ.0.0D0.AND.WSPX.EQ.0.0D0) GO TO 16
      IF(ISPIN.GT.1) SPNORB=.TRUE.
   15 IF (SPNORB) GO TO 18
C$============
   16 IF(WSTWO.GE.5) GO TO 18
C$============
      USPX=0.0D0
      WSPX=0.0D0
      RSPX=0.0D0
      ASP=0.0D0
C
C     INTERNAL  PARAMETERS
C
18     US=USX*CMU
       IF(MICR.EQ.1) GO TO 518
      RS=ABS(RSX)*ATHIRD
       IF(RSX.LT.0.0D0) RS=ABS(RSX)*AP3+RS
 518   WS=WSX*CMU
       IF(MICI.EQ.1) GO TO 618
      WD=WDX*CMU
      RI=ABS(RIX)*ATHIRD
       IF(RIX.LT.0.0D0) RI=RI+ABS(RIX)*AP3
618   RC=ABS(RCX)*ATHIRD
      IF(RCX.LT.0.0D0) RC=RC+ABS(RCX)*AP3
      IF(MICS.EQ.1) GO TO 718
C$$$$$$$$$$$$$$$$$$$$$$
      IF(WSTWO.GE.5) THEN
        RDV=ABS(RSPX)*ATHIRD
        IF(RSPX.LT.0.0D0) RDV= RDV+ ABS(RSPX)*AP3
        UDV= USPX*CMU
        ADV= ASP
        USP= 0.0D0
        WSP= 0.0D0
        RSP= 0.0D0
        GO TO 719
      ENDIF
C$$$$$$$$$$$$$$$$$$$$$$
      RSP=ABS(RSPX)*ATHIRD
      IF(RSPX.LT.0.0D0) RSP=RSP+ABS(RSPX)*AP3
      USP=USPX*CMU*4.0D0
      print *,"usp",usp,uspx,cmu
      WSP=WSPX*CMU*4.0D0
      print *,"wsp",wsp,wspx,cmu
      GO TO 719
  718 USP=USPX*CMU
      WSP=WSPX*CMU
      print *,"718"
  719 ECM=KSQ/CMU
C
C     RMAX,DR,MWV,VCCON
C
      IF(MICR.EQ.1.OR.MICI.EQ.1.OR.MICS.EQ.1) GO TO 1919
      IF(.NOT.NEWINT) GO TO 1818
      RMX0R=RMX(RS,AS,USX,1.D30)
C
C***************** HI-OPTIM ********** NOTES ************************
C
C      AMAX1 EXPLICITLY EDITED DMAX1 IN THIS ROUTINE TO
C      AVOID ID CONFLICT DUE TO VARIABLE NO''S OF ARGUMENTS.
C
C*****************************************************************
C
      RMX0I=RMX(RI,AI,DMAX1(WSX,4.0D0*WDX,WSX+4.0D0*WDX),1.D30)
      RMAXC=DMAX1(
     +RMX(RS,AS,USX,RMX0R),RMX(RI,AI,DMAX1(WSX,4.0D0*WDX,WSX+4.0
     $*WDX),RMX0I))
      IF (.NOT.NEWINT) GO TO 1818
      RMAX=RMAXIN
      print *,"rmax",rmax
      IF (RMAXIN.EQ.0.) RMAX=RMAXC*1.3D0
 1818 IF (RMAX.GT.RMAXC) GO TO 1919
        WRITE (6,1003) RMAXC
 1003 FORMAT ('17H0WARNING--RMAXC(=',F6.2,'10H ).GT.RMAX')
 1919 IF (.NOT.NEWINT) GO TO 40
      IF(MICR.EQ.0.AND.MICI.EQ.0.AND.MICS.EQ.0) GO TO 124
      RMAX=RMAXIN
124   DR=DRIN
```

```
      IF (DR.EQ.0.) DR=0.3D0/KWN
      SXTYDR=60.0D0*DR
      DRSQ=DR*DR
      DK=DRSQ*KSQ
      print *,"rmax",rmax
      NINT=MAX0(6,IFIX(RMAX/DR+.5D0))
      IF (NINT.LE.MINT-3) GO TO 19
         WRITE (6,1001) NINT
 1001 FORMAT ('47H0CALCULATION TERMINATED--NUMBER OF INTERVALS =',I3)
      ERROR=.TRUE.
      RETURN
   19 N1=NINT+1
      N3=NINT+3
      A3=FLOAT(N3)
      DO 20 I=N1,N3
      HIDRSQ(I)=0.0D0
      HRDRSQ(I)=0.0D0
      UDRSQ(I)=0.0D0
   20 WDRSQ(I)=0.0D0
      VCCONA=GAMMA*KWN*DRSQ
      VCCONB=VCCONA*2.0D0
      GO TO 50
C
C     COULOMB POTENTIAL
C
   40 IF (RC.EQ.RCSAV) GO TO 60
   50 RCSAV=RC
      VCCONC=VCCONA/RC
        RX=0.0D0
      DO 55 I=1,N3
      RX=RX+DR
      IF (RX.GT.RC) GO TO 52
      VCDRSQ(I)=VCCONC*(3.0D0-(RX*RX/(RC*RC)))
      GO TO 55
   52 VCDRSQ(I)=VCCONB/RX
   55 CONTINUE
      RMAX=FLOAT(NINT)*DR
      print *,"rmax",rmax
C
C     PHENOMENOLOGICAL SPIN-ORBIT POTENTIAL
C
      print *,"begin phenom"
   60 IF(SPNORB) GO TO 62
      IF(.NOT.NEWINT) GO TO 100
      print *,"60"
      DO 61 I=1,NINT
      HRDRSQ(I)=0.0D0
   61 HIDRSQ(I)=0.0D0
      PRINT "61"
      GO TO 100
   62 IF(MICS.EQ.0) GO TO 63
      IF(MICS.EQ.1) GO TO 83
   63 IF(.NOT.NEWINT) GO TO 65
      print *,"63"
      DO 64 I=1,NINT
      HRDRSQ(I)=0.0D0
   64 HIDRSQ(I)=0.0D0
      print *,"64"
      IF(NEWINT) GO TO 66
   65 IF(RSP.EQ.RSPSAV.AND.ASP.EQ.ASPSAV) GO TO 75
   66 IF(RSP.EQ.0.0D0.OR.ASP.EQ.0.0D0) GO TO 100
      PRINT *,"phenomil"
      RX=0.0D0
      BSO=EXP(DR/ASP)
      BBSO=BSO*EXP(-RSP/ASP)
C============ANTES ERA=================
C     IF(WSTWO.GT.2) GO TO 69
C======================================
      IF(WSTWO.EQ.3) GO TO 69
      DO 68 I=1,NINT
      RX=RX+DR
      IF(BBSO.GT.1.0D+20) BBSO=0.0D0
      TEMP=BBSO/(RX*ASP*(1.0D0+BBSO)*(1.0D0+BBSO))
      H(I)=TEMP*DRSQ
   68 BBSO=BBSO*BSO
      GO TO 76
  69  DO 70 I=1,NINT
      RX=RX+DR
      print *,"69"
      IF(BBSO.GT.1.0D+20) BBSO=0.0D0
      TEMP=1.0D0+BBSO
      H(I)=2.0D0*DRSQ*BBSO/(RX*ASP*TEMP*TEMP*TEMP)
  70  BBSO=BBSO*BSO
      GO TO 76
   75 IF (USP.EQ.USPSAV) GO TO 80
   76 USPSAV=USP
      print *,"76"
      DO 77 I=1,NINT
   77 HRDRSQ(I)=H(I)*USP
      print *,"77"
   80 IF (NEWINT) GO TO 81
      IF (WSP.EQ.WSPSAV.AND.RSP.EQ.RSPSAV.AND.ASP.EQ.ASPSAV) GO TO 100
   81 DO 82 I=1,NINT
   82 HIDRSQ(I)=H(I)*WSP
      print *,"82"
      WSPSAV=WSP
      RSPSAV=RSP
      ASPSAV=ASP
      GO TO 100
C
C      MICROSCOPIC SPIN-ORBIT POTENTIAL
C
   83 CONTINUE
   85 DO 86 I=1,NINT
      PRINT *,"micro"
      TEMP=DRSQ*H(I)
      HRDRSQ(I)=USP*TEMP
      print *,"85"
   86 HIDRSQ(I)=WSP*TEMP
      PRINT *,"micro2"
      USPSAV=USP
      WSPSAV=WSP
C
C      LMAX SET AND COULOMB WAVE FUNCTIONS CALCULATED
C
100   RHO=KWN*RMAX
      PRINT *,"100"
      AA=4.0D0*RHO*(RHO-2.0D0*GAMMA)
      BB=1.0D0+AA
      RLMAX=0.5D0*SQRT(BB)-0.5D0
      LMAX=IFIX(RLMAX)+15
      IF(LSET.EQ.1) LMAX=LMX
       IF(LMAX.GT.249) LMAX=249
      IF (.NOT.NEWINT) GO TO 150
      CALL COULOM  (ERROR,GAMMA,KWN,RMAX,DR,LMAX)
      IF (ERROR) RETURN
C
C      REAL NUCLEAR POTENTIALS
C
150   IF(MICR.EQ.0) GO TO 650
      IF(.NOT.NEWINT) GO TO 156
      DO 640 I=1,NINT
      UEXP(I)=DRSQ*UDRSQ(I)
640   CONTINUE
       USSAV=US
       GO TO 156
  650 IF (NEWINT) GO TO 151
      IF (RS.EQ.RSSAV.AND.AS.EQ.ASSAV) GO TO 155
  151 IF(RS.EQ.0.0D0.OR.AS.EQ.0.0D0) GO TO 160
      BR=EXP(DR/AS)
       BBR=BR*EXP(-RS/AS)
       IF(WSTWO.EQ.0) GO TO 251
C$=================NO ESTABA=========
       IF(WSTWO.EQ.4) GO TO 251
       IF(WSTWO.GE.5) GO TO 251
C$===================================
      DO 252 I=1,NINT
      UEXP (I)=0.!DRSQ/((1.0D0+BBR)*(1.0D0+BBR))
!     * +par1*bbr*bbr/(1.0D0+BBR)**4/as/as
!   print *, 'par1',par1
    print *,'aa'
  252 BBR=BBR*BR
      GOTO 253

  251 DO 152 I=1,NINT
      UEXP (I)=DRSQ*(us/(1.0D0+BBR)-16.*cmu*alpar*bbr*bbr/
     ^ ((1.0D0+BBR)**4))
c       print *,"deltaV correction applied"
c   print *,us,bbr,alpar
  152 BBR=BBR*BR
253   RSSAV=RS
      ASSAV=AS
      USSAV=US
      GO TO 156
  155 IF(US.EQ.USSAV.AND.WSTWO.LE.4) GO TO 500
C$========ANTES ERA=====
C$155 IF (US.EQ.USSAV) GO TO 160
C$=============
  156 DO 157 I=1,NINT
  157 UDRSQ(I)=UEXP(I)
      USSAV=US
C$$$$$$$$$$$$$
  160 IF(NEWINT) GO TO 400
      IF(RDV.EQ.RDVSAV.AND.ADV.EQ.ADVSAV) GO TO 401
  400 IF(RDV.EQ.0.0D0.AND.ADV.EQ.0.0D0) GO TO 500
      BCR= EXP(DR/ADV)
      BBCR=BCR*EXP(-RDV/ADV)
      DO 402 I= 1,NINT
       UEXP1(I)= BBCR
  402 BBCR=BBCR*BCR
      RDVSAV= RDV
      ADVSAV= ADV
  401 IF(WSTWO.EQ.7) THEN
      DO 405 I=1,NINT
  405 UDRSQ(I)= UDRSQ(I)+DRSQ*40.0D0*UDV/UEXP1(I)
      UDVSAV=UDV
      print *,"405"
      GO TO 500
      ENDIF
      DO 403 I=1,NINT
       TEMP= 1.0D0 +UEXP1(I)
  403 UDRSQ(I)= UDRSQ(I)+ DRSQ*40.0D0*UDV*UEXP1(I)/(TEMP*TEMP)
      UDVSAV= UDV
      print "403"
C$$$$$$$$$$$$$$$$$
C
C     IMAGINARY NUCLEAR POTENTIALS
C
  500 IF(MICI.EQ.0) GO TO 660
C$$$$$$$$ANTES ERA$$$$
C$160 IF(MICI.EQ.0) GO TO 660
C$$$$$$$$$$$$$$$$$$$$$$
      IF(.NOT.NEWINT) GO TO 3670
      DO 670 I=1,NINT
      WEXP (I)=DRSQ*WDRSQ(I)
670   CONTINUE
       WSSAV=1.0D0
```

```
3670   DO 3671 I=1,NINT
       WDRSQ(I)=WS*WEXP(I)
3671   CONTINUE
       WSSAV=WS
       GO TO 200
  660 IF (NEWINT) GO TO 161
      IF (RI.EQ.RISAV.AND.AI.EQ.AISAV) GO TO 165
  161 IF(RI.EQ.0.0D0.OR.AI.EQ.0.0D0) GO TO 180
      BI=EXP(DR/AI)
      BBI=BI*EXP(-RI/AI)
C$======================================
      BCI=EXP(DR/0.4)
      BCCI=BCI*EXP(-(AP3+ATHIRD)/0.4)
C$======================================
      DO 162 I=1,NINT
C$======================================
      WEXP1(I)=BCCI
      BCCI=BCI*BCCI
C$======================================
      WEXP(I)=BBI
  162 BBI=BBI*BI
      RISAV=RI
      AISAV=AI
      GO TO 166
  165 IF (WS.EQ.WSSAV.AND.WD.EQ.WDSAV) GO TO 180
  166 IF(WSTWO.GE.2.AND.WSTWO.LT.5) GO TO 168
C$$$$$$$$$$$$$$ANTES ERA$$$$$$$$$$$$$$$$$
C 166 IF(WSTWO.GE.2) GO TO 168
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      DO 167 I=1,NINT
      TEMP=1.0D0+WEXP(I)
  167 WDRSQ(I)=DRSQ*(WS/TEMP +4.0D0*WD*WEXP(I)/(TEMP*TEMP))
      GO TO 170
  168 CONTINUE
C$====================================
      IF(WSTWO.EQ.4.OR.WSTWO.EQ.6) GO TO 269
C$====================================
      DO 169 I=1,NINT
      TEMP=1.0D0+WEXP(I)
  169 WDRSQ(I)=DRSQ*(WS/(TEMP*TEMP)+ 8.0D0*WD*WEXP(I)/
     1        (TEMP*TEMP*TEMP))
      GO TO 170
C$===================================================================
  269 CONTINUE
      DO 369 I=1,NINT
      TEMP=1.0D0+WEXP(I)
      TEMP1=1.0D0+WEXP1(I)
  369 WDRSQ(I)=DRSQ*(WS/(TEMP1*TEMP1)+4.0D0*WD*WEXP(I)/(TEMP*TEMP))
C$===================================================================
  170 CONTINUE
      WSSAV=WS
      WDSAV=WD
C
C     POTENTIAL CUT-OFFS
C
180   IF(ICO.LT.1.OR.ICO.GT.3) GO TO 200
      M1=IFIX(RONE/DR+.5D0)
      M2=IFIX(RTWO/DR+.5D0)
      GO TO (5775,5777,5779),ICO
5775  DO 5776 I=M1,M2
5776  UDRSQ(I)=0.0D0
      GO TO 200
5777  DO 5778 I=M1,M2
5778  WDRSQ(I)=0.0D0
      GO TO 200
 5779 DO 5780 I=M1,M2
 5780 HRDRSQ(I)=0.0D0
C
C     NUCLEAR INTEGRATIONS INCORPORATING STARTING POINT ADVANCE
```

```
C
  200 CONTINUE
      N20=NINT-20
      N3M6=N3-6
      IF(LMN.LE.0.OR.LMN.GT.LMX) LMN=1
      DO 250 MWVS=LMN,LMX
      RMWVS=FLOAT(MWVS)
      FL=FLOAT(MWVS-1)
      ELTERM=FL*(FL+1.0D0)
      M1=SQRT((FL*FL+FL)/12.0D0)
      XPO=1.0D0/(2.0D0*FL+1.0D0)
      M2=((1.0D-16)**XPO)*(2.0D0*FL/(2.718*KWN*DR))
      M1=MAX0(M1,M2)+1
      MZ=MIN0(M1,N20)
      DO 249 MS=1,ISPIN
      J=0
      S=.5  [redacted]
      SZ=S+FLOAT(MS-ISPIN)
      EIGLS=((FL+SZ)*(FL+SZ+1.0D0)-ELTERM-S*(S+1.0D0))/2.0D0
      IF(MWVS.EQ.1) EIGLS=0.0D0
      IF(ISPIN.EQ.1.OR.LSTYPE.EQ.1) GO TO 210
      EIGLS=EIGLS/S
  210 CONTINUE
      UIB=0.0D0
      URB=0.0D0
      UIC=0.0D0
      URC=ETA1
      CRB=0.0D0
      IF(MWVS.EQ.2) CRB=2.0D0*ETA1
      CIB=0.0D0
      CIC=0.0D0
      CRC=0.0D0
      AM=0.0D0
      DO 248 M=1,N3
      AM=AM+1.0D0
      AMSQ=AM*AM
      IF(M.EQ.MZ) GO TO  230
      IF(M.LT.MZ) GO TO 248
C
C     P1 PREDICTOR
C
      URP=URC+URC-URB+CRC
      UIP=UIC+UIC-UIB+CIC
C
C     STEP BACK ONE
C
      CRA=CRB
      CIA=CIB
      CRB=CRC
      CIB=CIC
      URB=URC
      UIB=UIC
      URC=URP
      UIC=UIP
C
C     CALCULATE SECOND DERIVATIVES
C
 230  XR=-DK+VCDRSQ(M)-UDRSQ(M)-EIGLS*HRDRSQ(M)+ELTERM/AMSQ
c      print *,"eigls",eigls
      XI=-WDRSQ(M)-EIGLS*HIDRSQ(M)
c      print *,"imaginary comps",WDRSQ(M),EIGLS,HIDRSQ(M)
 231  CRC=XR*URC-XI*UIC
      CIC=XR*UIC+XI*URC
 240  IF(M.EQ.MZ) GO TO 247
C
C     C3 CORRECTOR
C
      URC=URC+R12*(CRC+CRA-CRB-CRB)
      UIC=UIC+R12*(CIC+CIA-CIB-CIB)
      CRCP=CRC
      CICP=CIC
C
C     BAYLIS PEEL CORRECTOR
C
      CRC=XR*URC-XI*UIC
      CIC=XR*UIC+XI*URC
      URC=URC+COR*(CRC-CRCP)
      UIC=UIC+COR*(CIC-CICP)
 247  IF(M.LT.N3M6) GO TO 248
      J=J+1
      UR(J)=URC
      UI(J)=UIC
 248  CONTINUE
      URP=(UR(7)-UR(1)+9.0D0*(UR(2)-UR(6))+45.0D0*(UR(5)-UR(3)))/SXTYDR
      UIP=(UI(7)-UI(1)+9.0D0*(UI(2)-UI(6))+45.0D0*(UI(5)-UI(3)))/SXTYDR
      U2=UR(4)**2+UI(4)**2
      DERR=(URP*UR(4)+UIP*UI(4))/U2
      DERI=(UIP*UR(4)-URP*UI(4))/U2
C
C          SCATTERING MATRIX
C
      SNR=FP(MWVS)-F(MWVS)*DERR+G(MWVS)*DERI
      SNI=GP(MWVS)-F(MWVS)*DERI-G(MWVS)*DERR
      SDR=F(MWVS)*DERR+G(MWVS)*DERI-FP(MWVS)
      SDI=GP(MWVS)+F(MWVS)*DERI-G(MWVS)*DERR
      SNN=SDR**2+SDI**2
      SR(MWVS,MS)=(SNR*SDR+SNI*SDI)/SNN
      SI(MWVS,MS)=(SNI*SDR-SNR*SDI)/SNN
  249 CONTINUE
      IF(LSET.EQ.1) GO TO 250
      DO 5247 I=1,ISPIN
      IF (ABS(SR(MWVS,I)).LE..99995D0.OR.ABS(SI(MWVS,I)).GE..00005D0)
     1 GO TO 5251
 5247 CONTINUE
      GO TO 300
5251   IF(MWVS.LT.LMAX) GO TO 250
       WRITE(6,5250)
 5250 FORMAT(' PARTIAL  WAVE SERIES TRUNCATED - INCREASE RMAX')
      GO TO 300
  250 CONTINUE
      MWVS=LMX
      IF(LSET.EQ.1) GO TO 300
       WRITE (6,1002)
 1002 FORMAT(' WARNING--SCATTERING MATRIX DOES NOT CONVERGE  ')
C
CC
  300 CONTINUE
       RETURN
      END
```