ADAS Subroutine h4born

```
C
       subroutine h4born( ixdim  , itdim   , isdim  , ndinfo ,
     &                    cameth ,
     &                    z0     , z       , zeff   ,
     &                    n1     , l1      , eb1    ,
     &                    n2     , l2      , eb2    ,
     &                    isp    , lp      , xjp    ,
     &                    ist1   , lt1     , xj1    , xjt1   ,
     &                    ist2   , lt2     , xj2    , xjt2   ,
     &                    neqv1  , fpc1    , neqv2  , fpc2   ,
     &                    aval   ,
     &                    xmax   , iext    , ijucys , jealfa ,
     &                    nshell , nc      , lc     , numel  , alfaa  ,
     &                    nx     , xa      , omga   ,
     &                    nt     , tea     , upsa   ,
     &                    cresol , ixtyp   , s12    , e12    ,
     &                    ninfo  , cinfoa
     &                  )
C-----------------------------------------------------------------------
C
C  **************** fortran77 program: h4born.for ********************
C
C  Purpose:  Calculation of Born cross-sections using numerical wave
C            functions.
C
C
C  Subroutine:
C
C  input : (i*4)  ixdim    = maximum dimension for X  array
C  input : (i*4)  itdim    = maximum dimension for Te array
C  input : (i*4)  isdim    = maximum dimension for shell vectors
C  input : (i*4)  ndinfo   = maximum number of information strings
C
C  input : (c*1)  cameth   = the tag distinguishing the type of
C                            analysis: a - Born, b- IP
C  input : (r*8)  z0       = nuclear charge
C  input : (r*8)  z        = ion charge
C  input : (r*8)  zeff     = ion charge +1
C  input : (i*4)  n1       = lower n-shell of transition
C  input : (i*4)  l1       = lower l-shell of transition
C  input : (r*8)  eb1      = binding energy (Ryd) in lower level
C  input : (i*4)  n2       = upper n-shell of transition
C  input : (i*4)  l2       = upper l-shell of transition
C  input : (r*8)  eb2      = binding energy (Ryd) in upper level
C  input : (i*4)  isp      = 2*Sp+1 for parent
C  input : (i*4)  lp       = Lp for parent
C  input : (r*8)  xjp      = Jp for parent (if 'ic' coupling)
C  input : (i*4)  ist1     = 2*S+1 for lower state
C  input : (i*4)  lt1      = L for lower state
C  input : (r*8)  xj1      = j for lower state
C  input : (r*8)  xjt1     = J for lower state
C  input : (i*4)  ist2     = 2*S'+1 for upper state
```

```
C  input : (i*4)  lt2      = L' for upper state
C  input : (r*8)  xj2      = j' for lower state
C  input : (r*8)  xjt2     = J' for upper state
C  input : (i*4)  neqv1    = no. of equiv. electrons for lower shell.
C  input : (r*8)  fpc1     = fract. parentage for lower state
C  input : (i*4)  neqv2    = no. of equiv. electrons for upper shell.
C  input : (r*8)  fpc2     = fract. parentage for upper state
C  input : (i*4)  aval     = A-value (sec-1) if dipole; else -ve
C  input : (i*4)  xmax     = range of numerical wave functions
C  input : (i*4)  iext     = 0 => calculate radial wave functions
C                          = 1 => read in radial wave functions
C  input : (i*4)  ijucys   = 0  => Slater potential form adopted
C                          = 1 => Jucys potential form adopted
C  input : (i*4)  jealfa   = 0 => fcf6 search for energy eigenvalue
C                          = 1 => fcf6 search for scaling parameters
C  input : (i*4)  nshell   = number of screening shells
C  input : (i*4)  nc()     =  n for each screening shell
C                             1st dim: screening shell index
C  input : (i*4)  lc()     = l for each screening shell
C                             1st dim: screening shell index
C  input : (i*4)  numel()  = iq for each screening shell
C                             1st dim: screening shell index
C  input : (r*8)  alfaa    = scaling factor for each screening shell
C                             1st dim: index for lower & upper state
C                             2nd dim: index over screening shells
C  input : (i*4)  nx       = number of incident electron energies
C  input : (r*8)  xa()     = threshold parameter values
C  input : (i*4)  nt       = number of electron temperatures
C  input : (r*8)  tea()    = electron temperatures (K)
C  input : (c*2)  cresol   = resolution level 'ls' or 'ic'

C  output: (r*8)  omga()   = omegas at input values of x-parameter
C  output: (r*8)  upsa()   = upsilon at onput values of Te
C  output: (i*4)  ixtyp    = transition type 1=dipole;2=non-dipole;
C                                             3=spin-change
C  output: (r*8)  s12      = line strength (dipole) otherwise zero
C  output: (r*8)  e12      = level energy difference (Ryd)
C  output: (i*4)  ninfo    = number of information strings
C  output  (c*90) cinfoa() = information strings
C                            1st dim: index number of strings
C
C        : (i*4)  icount    = general counter index
C        : (i*4)  iq        = general num. of equiv. electrons
C        : (i*4)  irept     = 0 => full wave function determination
C                           = 1 => use wave functions from previous case
C        : (i*4)  iswit     = 1 => no spin change
C                           = 2 => spin change
C        : (i*4)  j         = general integer variable
C        : (i*4)  j1        = general integer variable
C        : (i*4)  jalf1     = lower range of scaling parameters
C        : (i*4)  jalf2     = upper range of scaling parameters
C        : (i*4)  jh        = radial wave function tabulation size
C        : (i*4)  jsn       = -1 => Jucys potential form adopted
```

```
C                              = 0  => Slater potential form adopted
C         : (i*4)  lam         = multipole number
C         : (i*4)  n           = general principal quanum number
C         : (i*4)  nlam        = number of multipoles in anga vector
C         : (i*4)  nq          = number of equivalent electrons in initial
C                                state
C         : (r*8)  acc         = precision for eigenvalue search
C         : (r*8)  de          = transition energy (ryd)
C         : (r*8)  fpc         = fractional parentage coefficient
C         : (r*8)  ei          = incid. electron energy (ryd)
C         : (r*8)  eiu         = adj. incid. engy. for Cowan threshold (ryd)
C         : (r*8)  h           = step length for radial wave fn. tabulation
C         : (r*8)  omegb       = Born cross-section
C         : (r*8)  q           = number of equivalent electrons in initial
C                                state
C         : (r*8)  res         = general result variable
C         : (r*8)  t           = general real variable
C         : (r*8)  xlp         = Lp parent orbital angular momentum
C         : (r*8)  xl1         = l initial valence electron orbital
C         : (r*8)  xlt1        = L  total initial orbital angular momentum
C         : (r*8)  xl2         = l' final valence electron orbital
C         : (r*8)  xlt2        = L'  total initial final angular momentum
C         : (r*8)  xn1         = principal quantum number of initial shell
C         : (r*8)  xn2         = principal quantum number of final shell
C         : (r*8)  xnq         = number of equivalent electrons in initial
C         : (r*8)  xsp         = 2*Sp+1 parent spin angular momentum
C         : (r*8)  xst1        = 2*S+1  total initial spin anular momentum
C         : (r*8)  xst2        = 2*S'+1  total final spin anular momentum
C                                state
C         : (r*8)  z0          = nuclear charge
C         : (r*8)  z1          = ion charge + 1
C         : (r*8)  z           = ion charge
C         : (r*8)  zeff        = effective ion charge
C         : (r*8)  zz0         = -z0
C
C         : (i*4)  nlqs()      = 1000*n+100*l+iq for each screening shell
C                                 1st dim: screening shell index
C         : (i*4)  na()        = princ. qu. no. for val. elec.(initial & final)
C         : (i*4)  la()        = l qu. no. for val. elec.(initial & final)
C         : (i*4)  lama()      = multipole value
C
C         : (r*8)  alfa()      = scaling factor for each screening shell
C                                 1st dim: index over screening shells
C         : (r*8)  anga()      = angular factor for each multipole
C                                 1st dim: index over included multipoles
C         : (r*8)  ea()        = energy (ryd) for active electron.  NB -ve for
C                                a bound state
C                                 1st dim: index for initial & final state
C         : (r*8)  omega()     = collision strength for each multipole
C                                 1st dim: index over included multipoles
C         : (r*8)  qda()       = quantum defect for valence electron.
C                                 1st dim: index for initial & final state
C         : (r*8)  x0a()       = inner turning pt. for val. elec. wave fn.
```

```
C                              1st dim: index for initial & final state
C        : (r*8)  x1a()      = outer turning pt. for val. elec. wave fn.
C                              1st dim: index for initial & final state
C        : (r*8)  x2a()      = range for active elec. wave fn.
C                              1st dim: index for initial & final state
C
C
C  Routines:
C          routine    source    brief description
C          -------------------------------------------------------------
C          gamaf      adas      tabulates factorials
C          h4angf     adas      evaluates the Born angular parts
C          rdwbes     adas      evaluates the Born radial parts
C          i4unit     adas      fetch unit number for output of messages
C
C  Author:  H. P. Summers, University of Strathclyde
C           ja7.08
C           tel. 0141-548-4196
C
C  Date:   25/02/03
C
C  Update: HP Summers  21/05/04  Restructure and add calculation
C                                information strings to parameter output.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER           CAMETH
      CHARACTER*90        CINFOA(NDINFO)
      CHARACTER*2         CRESOL
      INTEGER             IEXT,         IJUCYS,       ISDIM,        ISP
      INTEGER             IST1,         IST2,         ITDIM,        IXDIM
      INTEGER             IXTYP,        JEALFA,       L1,           L2
      INTEGER             LC(ISDIM),    LP,           LT1,          LT2
      INTEGER             N1,           N2,           NC(ISDIM),    NDINFO
      INTEGER             NEQV1,        NEQV2,        NINFO,        NSHELL
      INTEGER             NT,           NUMEL(ISDIM),               NX
      REAL*8              ALFAA(2,ISDIM),             AVAL,         E12
      REAL*8              EB1,          EB2,          FPC1,         FPC2
      REAL*8              OMGA(IXDIM),  S12,          TEA(ITDIM)
      REAL*8              UPSA(ITDIM),  XA(IXDIM),    XJ1,          XJ2
      REAL*8              XJP,          XJT1,         XJT2,         XMAX
      REAL*8              Z,            Z0,           ZEFF
```