ADAS Subroutine hapecf

```
      subroutine hapecf( iunt15 , iunt40 , iunt11 , iunt11f , iunt35  ,
     &                   open15 , open40 , open11 , open11f , open35  ,
     &                   dsn35  ,
     &                   ndlev  , ndtrn  , ndtem  , ndden   , ndmet   ,
     &                   ndwvl  , ndpix  ,
     &                   nmet   , imetr  , nord   , iordr   ,
     &                   maxt   , teva   ,
     &                   maxd   , densa  ,
     &                   lpsel  , lzsel  , liosel ,
     &                   lhsel  , lrsel  , lisel  , lnsel   ,
     &                   iz     , iz0    , iz1    ,
     &                   npl    , bwno   ,
     &                   nplr   , npli   , npl3   ,
     &                   dsninc , dsnexp ,
     &                   titled , date   , user   ,
     &                   il     ,
     &                   ia     , cstrga , isa    , ila     , xja , wa ,
     &                   icnte  , icntr  , icnth  , icnti   ,
     &                   ietrn  ,
     &                   ie1a   , ie2a   , aa     ,
     &                   lnorm  ,
     &                   stckm  , stvr   , stvi   , stvh    ,
     &                   stvrm  , stvim  , stvhm  ,
     &                   ratpia , ratmia , stack  ,
     &                   lsseta , lss04a ,
     &                   nwvl   , npix   , wvmin  , wvmax   , avlt
     &                 )
c-----------------------------------------------------------------------
c
c  ******************** fortran77 subroutine: hapecf ******************
c
c  purpose:  to prepare pec, envelope feature f-pec, plt and plt-filter
c            passing files for diagnostic use.
c
c  calling program: adas810
c
c
c  subroutine:
c
c  input : (i*4)  iunt15  = output unit for adf15 pec results
c  input : (i*4)  iunt40  = output unit for adf40 fpec results
c  input : (i*4)  iunt11  = output unit for adf11 plt results
c  input : (i*4)  iunt11f = output unit for adf41 filtered plt results
c  input : (i*4)  iunt35  = input  unit for adf35 filter file
c
c  input : (i*4)  ndlev   = maximum number of levels allowed
c  input : (i*4)  ndtrn   = maximum number of transitions allowed
c  input : (i*4)  ndtem   = maximum number of temperatures allowed
c  input : (i*4)  ndden   = maximum number of densities allowed
c  input : (i*4)  ndmet   = maximum number of metastables allowed
c  input : (i*4)  ndwvl   = maximum number of wavelength intervals
c  input : (i*4)  ndpix   = maximum no. of pixels in a wvl. interval
```

```
c
c  input : (i*4)  nmet    = number of metastables levels: 1<=nmet<=ndmet
c  input : (i*4)  imetr() = index of metastable in complete level list
c  input : (i*4)  nord    = number of ordinary levels ('il' - 'nmet')
c  input : (i*4)  iordr() = index of ordinary levels in complete level
c                           list.
c
c  input : (i*4)  maxt    = number of input temperatures ( 1 -> 'ndtem')
c  input : (i*4)  maxd    = number of input densities ( 1 -> 'ndden')
c  input : (r*8)  teva()  = electron temperatures (units: ev)
c  input : (r*8)  densa() = electron densities  (units: cm-3)
c
c  input : (l*4)  lpsel    = .true.  => include proton collisions
c                          = .false. =>do not include proton collisions
c  input : (l*4)  lzsel    = .true.  => scale proton collisions with
c                                       plasma z effective'zeff'.
c                          = .false. => do not scale proton collisions
c                                       with plasma z effective 'zeff'.
c                           (only used if 'lpsel=.true.')
c  input   (l*4)  liosel  = .true.  => include ionisation rates
c                         = .false. => do not include ionisation rates
c                            for recom and  3-body
c  input   (l*4)  lhsel   = .true.  => include charge transfer from
c                                      neutral hydrogren.
c                         = .false. => do not include charge transfer
c                                      from neutral hydrogren.
c  input   (l*4)  lrsel   = .true.  => include free electron
c                                      recombination.
c                         = .false. => do not include free electron
c                                      recombination.
c  input   (l*4)  lisel   = .true.  => include electron impact
c                                      ionisation.
c                         = .false. => do not include free electron
c                                      recombination.
c  input : (l*4)  lnsel    = .true.  => include projected bundle-n data
c                                       from datafile if available
c                          = .false. => do not include projected
c                                       bundle-n data
c
c  input : (i*4)  iz      =  recombined ion charge read
c  input : (i*4)  iz0     =         nuclear charge read
c  input : (i*4)  iz1     = recombining ion charge read
c                           (note: iz1 should equal iz+1)
c
c  input : (i*4)  npl      = no. of metastables of(z+1) ion accessed
c                            by excited state ionisation in copase
c                            file with ionisation potentials given
c                            on the first data line
c  input : (r*8)  bwno    = ionisation potential (cm-1)
c
c  input : (i*4)  nplr     = no. of active metastables of (z+1) ion
c  input : (i*4)  npli     = no. of active metastables of (z-1) ion
c  input : (i*4)  npl3     = no. of active metastables of (z+1) ion with
```

```
c                              three-body recombination on.
c
c  input : (c*44) dsninc  = input copase data set name
c  input : (c*80) dsnexp  = input expansion file
c
c  input : (c*3)  titled  = element symbol.
c  input : (c*8)  date    = current date.
c  input : (c*30) user    = full name of author.
c
c  input : (i*4)  il      = number of energy levels
c
c  input : (i*4)  ia()    = energy level index number
c  input : (c*18) cstrga()= nomenclature/configuration for level 'ia()'
c  input : (i*4)  isa()   = multiplicity for level 'ia()'
c                           note: (isa-1)/2 = quantum number (s)
c  input : (i*4)  ila()   = quantum number (l) for level 'ia()'
c  input : (r*8)  xja()   = quantum number (j-value) for level 'ia()'
c                           note: (2*xja)+1 = statistical weight
c  input : (r*8)  wa()    = energy relative to level 1 (cm-1)
c                           dimension: level index
c
c
c  input : (i*4)  icnte   = number of electron impact transitions input
c  input : (i*4)  icntr   = number of free electron recombinations input
c  input : (i*4)  icnth   = no. of charge exchange recombinations input
c  input : (i*4)  icnti   = number of lower stage ionisations      input
c
c  input : (i*4)  ietrn() = electron impact transition:
c                           index values in main transition arrays which
c  input : (i*4)  ie1a()  = electron impact transition:
c                            lower energy level index
c  input : (i*4)  ie2a()  = electron impact transition:
c                            upper energy level index
c  input : (r*8)  aa()    = electron impact transition: a-value (sec-1)
c
c  input : (l*4)  lnorm   =.true.  => if nmet=1 then various
c                                     emissivity output files
c                                     normalised to stage tot.populatn.
c                                     (** norm type = t)
c                         =.false. => otherwise normalise to identified
c                                     metastable populations.
c                                      (** norm type = m)
c
c
c  input : (r*8) stckm(,,) = metastable populations stack
c                             1st dimension: metastable index
c                             2nd dimension: temperature index
c                             3rd dimension: density index
c  input : (r*4) stvr(,,,) = free electron recombination coefficients
c                             1st dimension: ordinary level index
c                             2nd dimension: temperature index
c                             3rd dimension: density index
c                             4th dimension: parent index
```

```
c  input : (r*4) stvi(,,,) = electron impact ionisation  coefficients
c                             1st dimension: ordinary level index
c                             2nd dimension: temperature index
c                             3rd dimension: density index
c                             4th dimension: parent index
c  input : (r*4) stvh(,,,) =  charge exchange coefficients
c                             1st dimension: ordinary level index
c                             2nd dimension: temperature index
c                             3rd dimension: density index
c  input : (r*8) stvrm(,,,)= metastable free electron recombination
c                            coefficients.
c                             1st dimension: metastable index
c                             2nd dimension: temperature index
c                             3rd dimension: density index
c                             4th dimension: parent index
c  input : (r*8) stvim(,,,)= metastable electron impact ionisation
c                            coefficients.
c                             1st dimension: metastable index
c                             2nd dimension: temperature index
c                             3rd dimension: density index
c                             4th dimension: parent index
c  input : (r*8) stvhm(,,,)= metastable charge exchange coefficients
c                             1st dimension: metastable index
c                             2nd dimension: temperature index
c                             3rd dimension: density index
c  input : (r*8) ratpia(,)   = ratio ( n(z+1)/n(z)  stage abundancies )
c                             1st dimension: temp/dens index
c                             2nd dimension: parent index
c  input : (r*8) ratmia(,)   = ratio ( n(z-1)/n(z)  stage abundancies )
c                             1st dimension: temp/dens index
c                             2nd dimension: parent index
c  input : (r*4) stack(,,,)= population dependence
c                             1st dimension: ordinary level index
c                             2nd dimension: metastable index
c                             3rd dimension: temperature index
c                             4th dimension: density index
c  input : (l*4) lsseta(,) = .true.  - met. ionis rate set in b8gets
c                            .false.- met. ionis rate not set in b8gets
c                             1st dimension: (z) ion metastable index
c                             2nd dimension: (z+1) ion metastable index
c  input : (l*4) lss04a(,) = .true. => ionis. rate set in adf04 file:
c                            .false.=> not set in adf04 file
c                             1st dim: level index
c                             2nd dim: parent metastable index
c
c  input : (i*4) nwvl      = number of wavelength intervals
c  input : (i*4) npix()    = number of pixels in each wvln. interval
c  input : (r*8) wvmin()   = minimum  wvln. (a) for each interval
c  input : (r*8) wvmax()   = maximum  wvln. (a) for each interval
c
c          (r*8) avlt      = lower limit of a-values for pec & f-pec
c
c          (i*4) notrn     = parameter = maximum number of transitions
```

```
c          (i*4) ndpec     = parameter = maximum number of pecs per
c                                        metastable for output
c          (i*4) metcnt    = counter of pecs for each metastable
c
c          (i*4) i4unit    = function (see routine selection below)
c
c          (i*4) i         = general use
c          (i*4) j         = general use
c          (i*4) k         = general use
c          (i*4) l         = general use
c
c          (r*8) dum1      = general use- dummy
c          (r*8) dum2      = general use- dummy
c          (r*8) dum3      = general use- dummy
c          (r*8) pec()     = renormalised pec
c                            1st dimension: temperature index
c
c routines:
c          ------------------------------------------------------------
c          hawvrg     adas      check for spectrum line in wvln.interval
c          hapixv     adas      doppler broaden line over pixel range
c          haout1     adas      writes plt and plt-filter output to files
c          b8norm     adas      perform  stage population normalisation
c          b8corp     adas      'fixes' low te problem in rec. data of pecs
c          i4unit     adas      fetch unit number for output of messages
c          xxordr     adas      sorts a real*8 array and its index array
c          xxeiam     adas      return the atomic mass of an element
c          xxmkrc     adas      make root connection vector
c          xxmkrp     adas      make root partition text lines for output
c          xxwcmt_15  adas      writes structured comments to adf15 dataset
c          xxwcmt_40  adas      writes structured comments to adf40 dataset
c
c author:  h. p. summers, university of strathclyde
c          tel: 0141-548-4196
c
c date:    24/04/02
c
c
c  version  : 1.1
c  date     : 24-02-2003
c  modified : H P Summers
c               - first version.
c
c  version  : 1.2
c  date     : 12-11-2003
c  modified : Martin O'Mullane
c               - trap plt and pltnfl for values below machine precision.
c               - increased number of transitions in line with 801/ifgpp.
c
c  version  : 1.3
c  date     : 05-12-2003
c  modified : Thomas Puetterich
c               - did not write f-pec file as per specification.
```

```
c
c  version  : 1.4
c  date     : 25-02-2004
c  modified : Martin O'Mullane
c              - increased number of transitions in line with 801/ifgpp.
c              - change behaviour of plt and filtered plt. no
c                limitations of wavelength or a-value to iunt11. an
c                adf35 filter is now an input and write plt filtered
c                by this to iunt11f.
c
c  version  : 1.5
c  date     : 26-05-2006
c  modified : Hugh Summers
c              - altered output on header lines for superstage
c                 compatibility.
c              - altered strategy for power ranking of emissivities
c              - altered comment lines for superstage compatibility
c                and field key reading of comments.
c
c  version  : 1.6
c  date     : 20-02-2007
c  modified : Martin O'Mullane
c              - Do not write comments to non-open units.
c              - Bring interactive version into line with
c                latest version of adf15 defintion (superstages).
c
c-----------------------------------------------------------------------
      CHARACTER*18        CSTRGA(NDLEV)
      CHARACTER*8         DATE
      CHARACTER*80        DSN35,       DSNEXP,      DSNINC
      CHARACTER*3         TITLED
      CHARACTER*30        USER
      INTEGER             IA(NDLEV),   ICNTE,       ICNTH,       ICNTI
      INTEGER             ICNTR,       IE1A(NDTRN), IE2A(NDTRN)
      INTEGER             IETRN(NDTRN),             IL
      INTEGER             ILA(NDLEV),  IMETR(NDMET)
      INTEGER             IORDR(NDLEV),             ISA(NDLEV),  IUNT11
      INTEGER             IUNT11F,     IUNT15,      IUNT35,      IUNT40
      INTEGER             IZ,          IZ0,         IZ1,         MAXD
      INTEGER             MAXT,        NDDEN,       NDLEV,       NDMET
      INTEGER             NDPIX,       NDTEM,       NDTRN,       NDWVL
      INTEGER             NMET,        NORD,        NPIX(NDWVL), NPL
      INTEGER             NPL3,        NPLI,        NPLR,        NWVL
      LOGICAL             LHSEL,       LIOSEL,      LISEL,       LNORM
      LOGICAL             LNSEL,       LPSEL,       LRSEL
      LOGICAL             LSS04A(NDLEV,NDMET),      LSSETA(NDMET,NDMET)
      LOGICAL             LZSEL,       OPEN11,      OPEN11F,     OPEN15
      LOGICAL             OPEN35,      OPEN40
      REAL*8              AA(NDTRN),   AVLT,        BWNO
      REAL*8              DENSA(NDDEN),             RATMIA(NDDEN,NDMET)
      REAL*8              RATPIA(NDDEN,NDMET)
      REAL                STACK(NDLEV,NDMET,NDTEM,NDDEN)
      REAL*8              STCKM(NDMET,NDTEM,NDDEN)
```

```
REAL              STVH(NDLEV,NDTEM,NDDEN,NDMET)
REAL*8            STVHM(NDMET,NDTEM,NDDEN,NDMET)
REAL              STVI(NDLEV,NDTEM,NDDEN,NDMET)
REAL*8            STVIM(NDMET,NDTEM,NDDEN,NDMET)
REAL              STVR(NDLEV,NDTEM,NDDEN,NDMET)
REAL*8            STVRM(NDMET,NDTEM,NDDEN,NDMET)
REAL*8            TEVA(NDTEM), WA(NDLEV),    WVMAX(NDWVL)
REAL*8            WVMIN(NDWVL),              XJA(NDLEV)
```