ADAS Subroutine bxcoef

```
       subroutine bxcoef(idlev  , idtrn  , idtem  , idden  , idmet ,
     &                   nmet   , imetr  , nord   , iordr  ,
     &                   maxt   , tine   , tinp   , tinh   , ifout ,
     &                   maxd   , dine   , dinp   , idout  ,
     &                   lpsel  , lzsel  , liosel ,
     &                   lhsel  , lrsel  , lisel  , lnsel  ,
     &                   iz     , iz0    , iz1    ,
     &                   npl    , bwno   , bwnoa  , prtwta ,
     &                   npla   , ipla   , zpla   , nplr   , npli  ,
     &                   il     , maxlev , xja    , wa     , zeff  ,
     &                   xia    , er     ,
     &                   icnte  , icntp  , icntr  , icnth  , icnti ,
     &                   icntl  , icnts  ,
     &                   ietrn  , iptrn  , irtrn  , ihtrn  , iitrn ,
     &                   iltrn  , istrn  ,
     &                   ie1a   , ie2a   , ip1a   , ip2a   , aa    ,
     &                   ia1a   , ia2a   , auga   ,
     &                   il1a   , il2a   , wvla   ,
     &                   is1a   , is2a   , lss04a ,
     &                   i1a    , i2a    ,
     &                   nv     , scef   , scom   ,
     &                   dsnexp , dsninc , iunt27 , open27 ,
     &                   stckm  , stvr   , stvi   , stvh   ,
     &                   stvrm  , stvim  , stvhm  , stack  ,
     &                   ltick
     &                  )


C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 SUBROUTINE: BXCOEF ********************
C
C  PURPOSE: Calculates collisional-radiative populations for series 8
C           codes.  Modification of bgcoef.for
C
C  CALLING PROGRAM: Series 2 population codes.
C
C
C  PARAMETERS:
C           (I*4)  ndlev   = parameter = max. number of levels allowed
C           (I*4)  ndtrn   = parameter = max. no. of transitions allowed
C           (I*4)  ndtem   = parameter = max. no. of temperatures allowed
C           (I*4)  ndden   = parameter = max. number of densities allowed
C           (I*4)  ndmet   = parameter = max. no. of metastables allowed
C
C
C  INPUT :  (I*4)  nmet    = number of metastables (1 <= nmet <= 'ndmet')
C  INPUT :  (I*4)  imetr() = index of metastable in complete level list
C                            (array size = 'ndmet' )
C  INPUT :  (I*4)  nord    = number of ordinary levels (1 <= nmet <= 'ndmet')
C  INPUT :  (I*4)  iordr() = index of metastable in complete level list
```

```
C                             (array size = 'ndmet' )
C
C  INPUT :  (I*4)  maxt    = number of input temperatures ( 1 -> 'ndtem')
C  INPUT :  (R*8)  tine()  = electron temperatures (units: see 'ifout')
C  INPUT :  (R*8)  tinp()  = proton temperatures   (units: see 'ifout')
C  INPUT :  (R*8)  tinh()  = neutral hydrogen temperatures
C  INPUT :  (I*4)  ifout   = 1 => input temperatures in kelvin
C                          = 2 => input temperatures in ev
C                          = 3 => input temperatures in reduced form
C
C  INPUT :  (I*4)  maxd    = number of input densities ( 1 -> 'ndden')
C  INPUT :  (R*8)  dine()  = electron densities  (units: see 'idout')
C  INPUT :  (R*8)  dinp()  = proton densities    (units: see 'idout')
C  INPUT :  (I*4)  idout   = 1 => input densities in cm-3
C                          = 2 => input densities in reduced form
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
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 bundle-n
C                                       data
C  INPUT :  (I*4)  iz      = recombined ion charge
C  INPUT :  (I*4)  iz0     = nuclear charge
C  INPUT :  (I*4)  iz1     = recombining ion charge
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) of lowest parent
C  INPUT :  (R*8)  bwnoa() = ionisation potential (cm-1) of parents
```

```
C  INPUT :  (R*8)  prtwta()= weight for parent associated with bwnoa()
C
C  INPUT :  (I*4)  npla()  = no. of parent/zeta contributions to ionis.
C                            of level
C  INPUT :  (I*4)  ipla(,) = parent index for contributions to ionis.
C                            of level
C                            1st dimension: parent index
C                            2nd dimension: level index
C  INPUT :  (I*4)  zpla(,) = eff. zeta param. for contributions to ionis.
C                            of level
C                            1st dimension: parent index
C                            2nd dimension: level index
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
C  INPUT :  (I*4)  il      = input data file: number of energy levels
C  INPUT :  (I*4)  maxlev  = highest index level in read transitions
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  INPUT :  (R*8)  zeff    = plasma z effective ( if 'lzsel' = .true.)
C                            (if 'lzsel' = .false. => 'zeff=1.0')
C
C  INPUT :  (R*8)  xia()   = energy relative to ion. pot. (rydbergs)
C                            dimension: level index
C           (R*8)  er()    = energy relative to level 1 (rydbergs)
C                            dimension: level index
C
C  INPUT :  (I*4)  icnte   = number of electron impact transitions input
C  INPUT :  (I*4)  icntp   = number of proton 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   = no. of ionisations to z input
C  INPUT :  (I*4)  icntl   = no. of satellite dr recombinations input
C  INPUT :  (I*4)  icnts   = no. of ionisations to z+1 input
C
C  INPUT :  (I*4)  ietrn() = electron impact transition:
C                            index values in main transition arrays which
C                            represent electron impact transitions.
C  INPUT :  (I*4)  iptrn() = proton impact transition:
C                            index values in main transition arrays which
C                            represent proton impact transitions.
C  INPUT :  (I*4)  irtrn() = free electron recombination:
C                            index values in main transition arrays which
C                            represent free electron recombinations.
C  INPUT :  (I*4)  ihtrn() = charge exchange recombination:
C                            index values in main transition arrays which
C                            represent charge exchange recombinations.
C  INPUT :  (I*4)  iitrn() = electron impact ionisation:
C                            index values in main transition arrays which
C                            represent ionisations from the lower stage
C  INPUT :  (I*4)  iltrn() = satellite dr recombination:
```

```
C                             index values in main transition arrays which
C                             represent satellite dr recombinations.
C  INPUT :  (I*4)  istrn() = electron impact ionisation:
C                             index values in main transition arrays which
C                             represent ionisations to upper stage ion.
C
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 :  (I*4)  ip1a()  = proton impact transition:
C                              lower energy level index
C  INPUT :  (I*4)  ip2a()  = proton impact transition:
C                              upper energy level index
C  INPUT :  (R*8)  aa()    = electron impact transition: a-value (sec-1)
C
C  INPUT :  (I*4)  ia1a()  = auger transition:
C                              parent energy level index
C  INPUT :  (I*4)  ia2a()  = auger transition:
C                              recombined ion energy level index
C  INPUT :  (R*8)  auga()  = auger transition: aug-value (sec-1)
C                              recombined ion energy level index
C  INPUT :  (I*4)  il1a()  = satellite dr transition:
C                              recomnining ion  index
C  INPUT :  (I*4)  il2a()  = satellite dr transition:
C                              recombined ion index
C  INPUT :  (R*8)  wvla()  = satellite dr transition: parent wvlgth.(a)
C                              dr satellite line index
C  INPUT :  (I*4)  is1a()  = ionising transition:
C                              ionised ion  index
C  INPUT :  (I*4)  is2a()  = ionising transition:
C                              ionising ion 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)  nv      = input data file: number of gamma/temperature
C                             pairs for a given transition.
C  INPUT :  (R*8)  scef()  = input data file: electron temperatures (k)
C  INPUT :  (R*8)  scom(,) = transition:
C                              gamma values    input :  (case ' ' & 'p')
C                              rate coefft. (cm3 sec-1) (case 'h' & 'r')
C                             1st dimension - temperature 'scef()'
C                             2nd dimension - transition number
C
C  INPUT :  (C*44) dsnexp  = expansion data set name
C  INPUT :  (C*44) dsninc  = input copase data set name (mvs dsn)
C  INPUT :  (I*4)  iunt27  = output unit for results from expansion routine
C  INPUT :  (L*4)  open27  = .true.  => file allocated to unit 7.
C                          = .false. => no file allocated to unit 7.
C
C
```

```
C
C  OUTPUT : (R*8)
C
C
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          b8getp     ADAS      Fetch expansion data
C          bxchkm     ADAS      Checks if transition exist to metastable
C          bxiord     ADAS      Sets  up ordinary level index.
C          bxrate     ADAS      Calculates exc. & de-exc. rate coeffts.
C          b8loss     ADAS      Calculates direct line power loss
C          b8rcom     ADAS      Establishes recombination rate coeffts.
C          bxmcra     ADAS      Constructs a-value matrix.
C          bxmcrc     ADAS      Constructs exc./de-exc. rate coef matrix
C          b8mcca     ADAS      Constructs whole rate matrix.
C          bxmcma     ADAS      Constructs ordinary level rate matrix.
C          bxstka     ADAS      Stack up ordinary pop. dependence on met
C          b8stkb     ADAS      Stack up recomb. contribution for ord.
C          bxstkc     ADAS      Stack up transition rate between mets.
C          b8stkd     ADAS      Stack up recomb rate for each met. level
C          b8stke     ADAS      Stack up recomb(+3-body) contri.for ord.
C          b8stkf     ADAS      Stack up recomb(+3-body) for each met.
C          bxmpop     ADAS      Calculate basic met. level populations.
C          b8stvm     ADAS      Calculate met. level recomb. coeffts.
C          xxtcon     ADAS      Converts ispf entered temps. to ev.
C          xxdcon     ADAS      Converts ispf entered dens. to cm-3.
C          xxrate     ADAS      Calculates exc. & de-exc. rate coeffts.
C                               For unit gamma value.
C          xxminv     ADAS      Inverts matrix and solves equations.
C
C
C
C  This is a subroutine version of adas208 without the search for ionisation
C  rates from adf07 files.  Modified original 'bgcoef.for' subroutine by
C  Martin O'Mullane to add extra variables to the returned parameter set.
C
C  AUTHOR:  H. P. Summers, University of Strathclyde
C          JA8.08
C          Tel. 0141-553-4196
C
C  DATE:    20/04/02
C
C  UPDATE:
C
C  VERSION  : 1.1
C  DATE     : 20-01-2003
C  MODIFIED : Martin O'Mullane
C             - based on bgcoef v 1.1.
C             - Remove calculation of populations and lsseta as input.
C
```

```
C  VERSION  : 1.2
C  DATE     : 29-05-2003
C  MODIFIED : Martin O'Mullane
C             - Increase NDLEV to 1200.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER*80        DSNEXP,      DSNINC
      INTEGER             I1A(NDTRN),  I2A(NDTRN),  IA1A(NDTRN)
      INTEGER             IA2A(NDTRN), ICNTE,       ICNTH,       ICNTI
      INTEGER             ICNTL,       ICNTP,       ICNTR,       ICNTS
      INTEGER             IDDEN,       IDLEV,       IDMET,       IDOUT
      INTEGER             IDTEM,       IDTRN,       IE1A(NDTRN)
      INTEGER             IE2A(NDTRN), IETRN(NDTRN),             IFOUT
      INTEGER             IHTRN(NDTRN),             IITRN(NDTRN)
      INTEGER             IL,          IL1A(NDLEV), IL2A(NDLEV)
      INTEGER             ILTRN(NDTRN),             IMETR(NDMET)
      INTEGER             IORDR(NDLEV),             IP1A(NDTRN)
      INTEGER             IP2A(NDTRN), IPLA(NDMET,NDLEV)
      INTEGER             IPTRN(NDTRN),             IRTRN(NDTRN)
      INTEGER             IS1A(NDLEV), IS2A(NDLEV), ISTRN(NDTRN)
      INTEGER             IUNT27,      IZ,          IZ0,         IZ1
      INTEGER             MAXD,        MAXLEV,      MAXT,        NMET
      INTEGER             NORD,        NPL,         NPLA(NDLEV), NPLI
      INTEGER             NPLR,        NV
      LOGICAL             LHSEL,       LIOSEL,      LISEL,       LNSEL
      LOGICAL             LPSEL,       LRSEL,       LSS04A(IDLEV,IDMET)
      LOGICAL             LTICK,       LZSEL,       OPEN27
      REAL*8              AA(NDTRN),   AUGA(NDTRN), BWNO
      REAL*8              BWNOA(NDMET),             DINE(NDDEN)
      REAL*8              DINP(NDDEN), ER(NDLEV),   PRTWTA(NDMET)
      REAL*8              SCEF(14),    SCOM(14,NDTRN)
      REAL                STACK(IDLEV,IDMET,IDTEM,IDDEN)
      REAL*8              STCKM(IDMET,IDTEM,IDDEN)
      REAL                STVH(IDLEV,IDTEM,IDDEN,IDMET)
      REAL*8              STVHM(IDMET,IDTEM,IDDEN,IDMET)
      REAL                STVI(IDLEV,IDTEM,IDDEN,IDMET)
      REAL*8              STVIM(IDMET,IDTEM,IDDEN,IDMET)
      REAL                STVR(IDLEV,IDTEM,IDDEN,IDMET)
      REAL*8              STVRM(IDMET,IDTEM,IDDEN,IDMET)
      REAL*8              TINE(NDTEM), TINH(NDTEM), TINP(NDTEM)
      REAL*8              WA(NDLEV),   WVLA(NDLEV), XIA(NDLEV)
      REAL*8              XJA(NDLEV),  ZEFF,        ZPLA(NDMET,NDLEV)
```