ADAS Subroutine cxdata

```
       SUBROUTINE CXDATA( IUNIT  , MXNENG , MXNSHL , TITLED ,
     &                    SYMBR  , SYMBD  , IZR    , IZD    ,
     &                    INDD   , NENRGY , NMIN   , NMAX   ,
     &                    LPARMS , LSETL  , LSETM  , ENRGYA ,
     &                    ALPHAA , LFORMA , XLCUTA , PL2A   ,
     &                    PL3A   , SIGTA  , SIGNA  , SIGLA  ,
     &                    SIGMA
     &                  )
C
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: CXDATA *********************
C
C  PURPOSE:  TO FETCH DATA FROM INPUT DATA SET OF TYPE ADF01.
C
C  CALLING PROGRAM: ADAS301/ADAS306/ADAS307/ADAS308/ADAS309
C
C  DATA:
C
C           THE UNITS USED IN THE DATA FILE ARE TAKEN AS FOLLOWS:
C
C           COLLISION ENERGIES  : KEV/AMU
C           ALPHA               :
C           TOTAL XSECTS.       : CM2
C           N-SHELL XSECTS.     : CM2
C           NL-SHELL DATA       : CM2
C           NLM-SHELL DATA      : CM2
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT     = UNIT TO WHICH INPUT FILE IS ALLOCATED.
C  INPUT : (I*4)  MXNENG    = MAXIMUM NO. OF ENERGIES.
C  INPUT : (I*4)  MXNSHL    = MAXIMUM NO. OF N SHELLS.
C
C  OUTPUT: (C*80) TITLED    = NOT SET - TITLE FOR DATA SOURCE.
C  OUTPUT: (C*2)  SYMBR     = READ - RECEIVER ION ELEMENT SYMBOL.
C  OUTPUT: (C*2)  SYMBD     = READ - DONOR ION ELMENT SYMBOL.
C  OUTPUT: (I*4)  IZR       = READ - ION CHARGE OF RECEIVER.
C  OUTPUT: (I*4)  IZD       = READ - ION CHARGE OF DONOR.
C  OUTPUT: (I*4)  INDD      = READ - DONOR STATE INDEX.
C  OUTPUT: (I*4)  NENRGY    = NUMBER OF ENERGIES READ.
C  OUTPUT: (I*4)  NMIN      = LOWEST N-SHELL FOR WHICH DATA READ.
C  OUTPUT: (I*4)  NMAX      = HIGHEST N-SHELL FOR WHICH DATA READ.
C  OUTPUT: (L*4)  LPARMS    = FLAGS IF L-SPLITTING PARAMETERS PRESENT.
C                             .TRUE.  => L-SPLITTING PARAMETERS PRESENT.
C                             .FALSE  => L-SPLITTING PARAMETERS ABSENT.
C  OUTPUT: (L*4)  LSETL     = FLAGS IF L-RESOLVED DATA PRESENT.
C                             .TRUE.  => L-RESOLVED DATA PRESENT.
C                             .FALSE  => L-RESOLVED DATA ABSENT.
C  OUTPUT: (L*4)  LSETM     = FLAGS IF M-RESOLVED DATA PRESENT.
```

```
C                              .TRUE.  => M-RESOLVED DATA PRESENT.
C                              .FALSE  => M-RESOLVED DATA ABSENT.
C
C  OUTPUT: (R*8)  ENRGYA()  = READ - COLLISION ENERGIES.
C                              UNITS: EV/AMU (READ AS KEV/AMU)
C                              DIMENSION: ENERGY INDEX
C  OUTPUT: (R*8)  ALPHAA()  = READ - EXTRAPOLATION PARAMETER ALPHA.
C                              DIMENSION: ENERGY INDEX
C  OUTPUT: (I*4)  LFORMA()  = READ - PARAMETERS FOR CALCULATING L-RES
C                                    X-SEC.
C                              DIMENSION: ENERGY INDEX
C  OUTPUT: (R*8)  XLCUTA()  = READ - PARAMETERS FOR CALCULATING L-RES
C                                    X-SEC.
C                              DIMENSION: ENERGY INDEX
C  OUTPUT: (R*8)  PL2A()    = READ - PARAMETERS FOR CALCULATING L-RES
C                                    X-SEC.
C                              DIMENSION: ENERGY INDEX
C  OUTPUT: (R*8)  PL3A()    = READ - PARAMETERS FOR CALCULATING L-RES
C                                    X-SEC.
C                              DIMENSION: ENERGY INDEX
C  OUTPUT: (R*8)  SIGTA()   = READ - TOTAL CHARGE EXCHANGE
C                                    CROSS-SECTION.
C                              UNITS: CM2
C                              DIMENSION: ENERGY INDEX
C  OUTPUT: (R*8)  SIGNA(,)  = READ - N-RESOLVED CHARGE EXCHANGE
C                                    CROSS-SECTIONS.
C                              UNITS: CM2
C                              1ST DIMENSION: ENERGY INDEX
C                              2ND DIMENSION: N-SHELL
C  OUTPUT: (R*8)  SIGLA(,)  = READ - L-RESOLVED CHARGE EXCHANGE
C                                    CROSS-SECTIONS.
C                              UNITS: CM2
C                              1ST DIMENSION: ENERGY INDEX
C                              2ND DIMENSION: INDEXED BY I4IDFL(N,L)
C  OUTPUT: (R*8)  SIGMA(,)  = READ - M-RESOLVED CHARGE EXCHANGE
C                                    CROSS-SECTIONS.
C                              UNITS: CM2
C                              1ST DIMENSION: ENERGY INDEX
C                              2ND DIMENSION: INDEXED BY I4IDFM(N,L,M)
C                                             WITH M >= 0 ONLY
C
C    (R*8)  ZEROST   = PARAMETER = EFFECTIVE SHIFT APPLIED TO
C  CROSS-SECTION VALUES TO AVOID
C  ZERO VALUES (WILL NOT AFFECT
C  ANY VALUES WHICH ARE GREATER
C  THAN AROUND 1.0E+15*ZEROSHFT -
C  i.e. 1.0E-25.)
C
C          (I*4)  OLDMIN   = PREVIOUS VALUE READ FOR NMIN.
C          (I*4)  OLDMAX   = PREVIOUS VALUE READ FOR NMAX.
C          (I*4)  IBLK     = CURRENT DATA BLOCK.
C          (I*4)  IVALUE   = USED TO PARSE FOR END OF DATA FLAG (-1).
C          (I*4)  N        = N QUANTUM NUMBER.
```

```
C          (I*4)  L        = L QUANTUM NUMBER.
C          (I*4)  M        = M QUANTUM NUMBER.
C          (I*4)  I        = LOOP COUNTER.
C          (I*4)  J        = LOOP COUNTER.
C          (I*4)  IERR     = ERROR RETURN CODE.
C          (C*2)  CIZR     = ION CHARGE OF RECEIVER.
C          (C*2)  CIZD     = ION CHARGE OF DONOR.
C          (C*1)  INDD     = DONOR STATE INDEX.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4FCTN     ADAS      RETURNS CHARACTER STRING AS AN INTEGER.
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX FROM QUANTUM
C                               NUMBERS N AND L.
C          I4IDFM     ADAS      RETURNS UNIQUE INDEX FROM QUANTUM
C                               NUMBERS N, L AND M.
C          XXIDTL     ADAS      INVERSE OF I4IDFL. RETURNS QUANTUM
C                               NUMBERS N AND L FROM INDEX.
C          XXIDTM     ADAS      INVERSE OF I4IDFM. RETURNS QUANTUM
C                               NUMBERS N, L AND M FROM INDEX.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    21/09/93
C
C UPDATE:  18/10/93 - J NASH    - ADAS91:
C          UPDATED TO READ L-SPLITTING PARAMETERS IF PRESENT IN DATASET.
C
C UPDATE:  01/05/95 - Tim Hammond - IDLADAS:
C    UNIX port.
C
C UPDATE:  16/05/95 - Tim Hammond - IDLADAS:
C    ADDED AND APPLIED ZEROST PARAMETER => EFFECTIVE ZERO FOR
C          CROSS-SECTIONS (CODING DONE BY PAUL BRIDEN).
C
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER*2         SYMBD,       SYMBR
      CHARACTER*80        TITLED
      INTEGER             INDD,        IUNIT,       IZD,         IZR
      INTEGER             LFORMA(MXNENG),           MXNENG,      MXNSHL
      INTEGER             NENRGY,      NMAX,        NMIN
      LOGICAL             LPARMS,      LSETL,       LSETM
      REAL*8              ALPHAA(MXNENG),           ENRGYA(MXNENG)
      REAL*8              PL2A(MXNENG),             PL3A(MXNENG)
      REAL*8              SIGLA(MXNENG,(MXNSHL*(MXNSHL+1))/2)
      REAL*8              SIGMA(MXNENG,(MXNSHL*(MXNSHL+1)*(MXNSHL+2))/6)
```

```
REAL*8              SIGNA(MXNENG,MXNSHL),     SIGTA(MXNENG)
REAL*8              XLCUTA(MXNENG)
```