ADASLIB FORTRAN

For subroutine naming, codes series 1-7 are labelled alphabetically a-g. A subroutine associated with series 1 begins with the letter ‘a’, a subroutine associated with series 2 begins with letter ‘b’ and so on. A subroutine associated with a particular member of a code series has the code number as the second character of the subroutine name. Thus the name of a subroutine associated with the code ADAS403 begins with the letters ‘d3’ (e.g. ‘d3spf0’). The name of a general subroutine from a code series but not restricted to a specific code has ‘x’ as the second character of its name (e.g. ‘bxdata’). A general subroutine, not restricted to a single code series begins with the letters ‘xx’ (e.g. ‘xxuid’). General functions are preceded by ‘I4’, ‘R8’, ‘C’ depending on the type of function value returned.

FORTRAN subroutine names have the terminator ‘.for’. FORTRAN subroutine source code is not accessible, but the headers, including call parameters, description of the routine and all variables and declaration statements are provided in this document.

The FORTRAN scannable libraries are located as follow:

/…/adas/idl_adas/fortran/adaslib.a	‘XX’,’I4’, ‘R8’ and ‘C’ general fortran subroutines and functions
/…/adas/idl_adas/fortran/adas1xx.a	‘AX’, ‘A1’, ‘A2’, … fortran subroutines and functions
/…/adas/idl_adas/fortran/adas2xx.a	‘BX’, ‘B1’, ‘B2’, … fortran subroutines and functions
/…/adas/idl_adas/fortran/adas3xx.a	‘CX’, ‘C1’, ‘C2’, … fortran subroutines and functions
/…/adas/idl_adas/fortran/adas4xx.a	‘DX’, ‘D1’, ‘D2’, … fortran subroutines and functions
/…/adas/idl_adas/fortran/adas5xx.a	‘EX’, ‘E1’, ‘E2’, … fortran subroutines and functions
/…/adas/idl_adas/fortran/adas6xx.a	‘FX’, ‘F1’, ‘F2’, … fortran subroutines and functions
/…/adas/idl_adas/fortran/adas7xx.a	‘GX’, ‘G1’, ‘G2’, … fortran subroutines and functions

CSTGRP

```
      FUNCTION CSTGRP(CHR)
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: CSTGRP  *******************
C
C  PURPOSE: RETURNS TERM OF ORBITAL GIVEN IN THE EISSNER SINGLE
C           HEXADECIMAL CHARACTER FORM
C
C
C  INPUT :  (C*1)   CHR    = HEX. ORBITAL CHARACTER (EG. 'A')
C
C  OUTPUT:  (C*2)   CSTGRP = ORBITAL TERM   (E.G. '4F')
C
C           (I*4)   I      = GENERAL INDEX
C
C  SUBROUTINES:  NONE
C  AUTHOR:  W.J. DICKSON, JET
C
C  DATE:    20/10/89
C
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 22-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
      INTEGER      I
C-----------------------------------------------------------------------
      CHARACTER*1  CHR    , CHRA(15)
      CHARACTER*2  CSTGRP , TGRPA(15)
C-----------------------------------------------------------------------
```

I4EIZ0

```
      FUNCTION I4EIZ0 ( ESYM )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C *************** FORTRAN77 INTEGER*4 FUNCTION: I4EIZ0 *****************
C
C PURPOSE: TO RETURN THE NUCLEAR CHARGE FOR THE ELEMENT SYMBOL ESYM
C          (INTEGER*4 FUNCTION VERSION OF 'XXEIZ0')
C
C CALLING PROGRAM: GENERAL USE
C
C FUNCTION:
C
C          (I*4)  I4EIZ0  = FUNCTION NAME -
C                           ELEMENT NUCLEAR CHARGE
C          (C*2)  ESYM    = SYMBOL OF ELEMENT WITH NUCLEAR CHARGE I4EIZ0
C
C          (I*4)  NSYM    = PARAMETER = NUMBER OF SYMBOLS LISTED
C
C          (I*4)  I       = GENERAL ARRAY USE
C
C          (C*2)  SYMBOL()= SYMBOLS OF FIRST 'NSYM' ELEMENTS.
C                           ARRAY DIMENSION => NUCLEAR CHARGE
C
C NOTES:    IF SYMBOL IS NOT RECOGNISED, I.E.NOT IN Z0 RANGE 1 & 'NSYM',
C           THEN THE INTEGER 'I4EIZ0' IS RETURNED AS ZERO.
C
C ROUTINES: NONE
C
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     13/02/91
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER    I4EIZ0    , NSYM
C-----------------------------------------------------------------------
      PARAMETER( NSYM = 50 )
C-----------------------------------------------------------------------
      INTEGER    I
C-----------------------------------------------------------------------
      CHARACTER  ESYM*2    , SYMBOL(NSYM)*2
C-----------------------------------------------------------------------
```

I4FCTN

```
      FUNCTION I4FCTN( STR , IABT )
      IMPLICIT NONE
```

```
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 INTEGER*4 FUNCTION: I4FCTN ****************
C
C  FUNCTION: TO CONVERT AN INTEGER NUMBER STORED IN THE STRING 'STR'
C            INTO A INTEGER*4 VARIABLE, USING INTERNAL READ.
C            INTIALLY THE PROGRAM CHECKS TO SEE IF THE NUMBER IS OF A
C            VALID FORM.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C          (I*4)   I4FCTN  = FUNCTION NAME
C          (C*(*)) STR     = STRING CONTAINING SINGLE FLOATING POINT NO.
C          (I*4)   IABT    = RETURN CODE:
C                                0 => NO ERROR
C                                1 => ERROR (A VALUE 'I4FCTN=0' WILL BE
C                                           RETURNED).
C
C          (C*1)   CH0      = PARAMETER = '0'
C          (C*1)   CH9      = PARAMETER = '9'
C          (C*1)   BLANK    = PARAMETER = ' '
C          (C*1)   CPLUS    = PARAMETER = '+'
C          (C*1)   CMINUS   = PARAMETER = '-'
C
C          (I*4)   ILEN     = LENGTH OF 'STR' STRING IN BYTES
C          (I*4)   ILAST    = POSITION OF LAST BYTE OF IDENTIFIED NUMBER
C          (I*4)   I1       = STARTING BYTE IN 'STR' OF NUMBER
C                             INCLUDING SIGN IF PRESENT
C          (I*4)   IS       = 0 => NUMBER HAS NO SIGN
C                             1 => NUMBER HAS A SIGN
C          (I*4)   ICH0     = ICHAR('0')
C          (I*4)   ICH9     = ICHAR('9')
C          (I*4)   ISTR     = ICHAR(CURRENT BYTE POSITION IN 'STR')
C          (I*4)   I        = GENERAL USE
C
C          (L*4)   LFOUND   = .TRUE.  => ALL OF THE INPUT NUMBER BYTES
C                                        HAVE BEEN ASSESSED.
C                             .FALSE. => INPUT NUMBER BYTES STILL BEING
C                                        ASSESSED.
C          (L*4)   LSTART   = .TRUE.  => THE FIRST DIGIT HAS BEEN FOUND
C                             .FALSE. => THE FIRST DIGIT HAS NOT YET
C                                        BEEN REACHED.
C
C          (C*5)   CFORM5   = FORMAT FOR INTERNAL READING OF INTEGER
C
C
C NOTE:     AN ERROR WILL OCCUR (IABT=1) IF THERE IS MORE THAN ONE
C           NUMBER OCCURING IN THE STRING 'STR()'
C
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/37
C           JET EXT. 2520
C
C DATE:     11/07/90
C
C UPDATE:   11/02/92 - PE BRIDEN: BLANKS NOW ALLOWED BETWEEN SIGN AND
C                                 FIRST DIGIT. LSTART VARIABLE ADDED.
C                                 VARIABLE I2 REMOVED.
C                                 + SOME MINOR RECODING - (IF STRING
C                                 ENTERED IS BLANK IABT IS NOW SET TO 1)
C
C UPDATE:   16/08/93 - PE BRIDEN: CORRECTED BUG TO ALLOW BLANKS BETWEEN
C                                 SIGN AND FIRST DIGIT (SEE ABOVE).
C                                 1) ILAST VARIABLE ADDED.
C                                 2) FORMATTED READ USED INSTEAD OF *
C                                    WHEN CONVERTING IDENTIFIED INTEGER
C                                    USING THE INTERNAL READ. (THIS
C                                    RESTRICTS IDENTIFIED NUMBER TO BE
C                                    < 100 BYTES IN LENGTH!)
C                                 3) EXCLUDE TRAILING BLANKS IN THE
C                                    INTERNAL READING OF THE INTEGER
C                                    I.E. STR(I1:ILAST) INSTEAD OF
C                                         STR(I1:ILEN)
C
C UPDATE:   07/03/95 - PE BRIDEN: INSTEAD OF USING FORMAT SPECIFIER I99
C                                 WHEN INTERNALLY READING THE INTEGER
C                                 CREATE THE APPROPRIATE SPECIFIER
C                                 WITHIN CFORM5 AND USE THIS.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      CHARACTER  CH0*1   , CH9*1   , BLANK*1   , CPLUS*1   , CMINUS*1
C-----------------------------------------------------------------------
      PARAMETER( CH0='0', CH9='9', BLANK=' ', CPLUS='+', CMINUS='-' )
C-----------------------------------------------------------------------
      CHARACTER  STR*(*)
C-----------------------------------------------------------------------
      INTEGER    I4FCTN  , IABT
      INTEGER    I1      , IS      , ILEN      , ILAST     ,
     &           ICH0    , ICH9    , ISTR      , I
C-----------------------------------------------------------------------
      LOGICAL    LSTART  , LFOUND
C-----------------------------------------------------------------------
      CHARACTER  CFORM5*5
```

```
C-----------------------------------------------------------------------
      DATA       CFORM5 / '(I??)' /
C-----------------------------------------------------------------------
```

I4IDFL

```
      FUNCTION I4IDFL( N , L ) 0
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 INTEGER*4 FUNCTION: I4INDL ****************
C
C  FUNCTION: RETURNS A UNIQUE INDEX NUMBER BASED ON THE VALUE OF THE
C            N AND L QUANTUM NUMBERS PASSED TO IT. THE INDEX IS USED TO
C            REFERENCE ARRAYS CONTAINING DATA DEPENDENT ON THE N AND L
C            QUANTUM NUMBERS.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C  FUNC:   (I*4)   I4IDFL  = INDEX
C
C  INPUT:  (I*4)   N       = N QUANTUM NUMBER.
C  INPUT:  (I*4)   L       = L QUANTUM NUMBER.
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     10/09/93
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER    I4IDFL
C-----------------------------------------------------------------------
      INTEGER    N  , L
C-----------------------------------------------------------------------
```

I4IDFM

```
      FUNCTION I4IDFM( N , L , M )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 INTEGER*4 FUNCTION: I4INDL ****************
C
C  FUNCTION: RETURNS A UNIQUE INDEX NUMBER BASED ON THE VALUE OF THE
C            N, L AND M QUANTUM NUMBERS PASSED TO IT. THE INDEX IS USED
C            TO REFERENCE ARRAYS CONTAINING DATA DEPENDENT ON THE
C            N, L AND M QUANTUM NUMBERS.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C  FUNC:   (I*4)   I4IDFM  = INDEX
C
C  INPUT:  (I*4)   N       = N QUANTUM NUMBER.
C  INPUT:  (I*4)   L       = L QUANTUM NUMBER.
C  INPUT:  (I*4)   M       = M QUANTUM NUMBER.
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     10/09/93
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER    I4IDFM
C-----------------------------------------------------------------------
      INTEGER    N  , L  , M
C-----------------------------------------------------------------------
```

I4IDFM

```
      FUNCTION I4IDFM( N , L , M )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 INTEGER*4 FUNCTION: I4INDL ****************
C
C  FUNCTION: RETURNS A UNIQUE INDEX NUMBER BASED ON THE VALUE OF THE
C            N, L AND M QUANTUM NUMBERS PASSED TO IT. THE INDEX IS USED
C            TO REFERENCE ARRAYS CONTAINING DATA DEPENDENT ON THE
C            N, L AND M QUANTUM NUMBERS.
C
C  CALLING PROGRAM: GENERAL USE
```

```
C
C  FUNCTION:
C
C  FUNC:   (I*4)   I4IDFM  = INDEX
C
C  INPUT:  (I*4)   N       = N QUANTUM NUMBER.
C  INPUT:  (I*4)   L       = L QUANTUM NUMBER.
C  INPUT:  (I*4)   M       = M QUANTUM NUMBER.
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     10/09/93
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER    I4IDFM
C-----------------------------------------------------------------------
      INTEGER    N  , L  , M
C-----------------------------------------------------------------------
```

I4IDLI

```
      FUNCTION I4IDLI( NF , LI , LF )
      IMPLICIT NONE
C
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 INTEGER*4 FUNCTION: I4IDLI ****************
C
C  FUNCTION: RETURNS THE INDEX NUMBER OF THE PREDICTED SPETRUM LINE
C            TABLES GIVEN THE ORBITAL QUANTUM NUMBER OF THE INTIAL STATE
C            AND THE PRINCIPAL AND ORBITAL QUANTUM NUMBERS OF THE FINAL
C            STATE.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C  FUNC:   (I*4)   I4IDLI  = INDEX
C
C  INPUT:  (I*4)   NF      = PRINCIAPL QUANTUM NUMBER OF FINAL STATE.
C  INPUT:  (I*4)   LI      = ORBITAL QUANTUM NUMBER OF INITIAL STATE.
C  INPUT:  (I*4)   LF      = ORBITAL QUANTUM NUMBER OF FINAL STATE.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    14/10/93
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER    I4IDLI
C-----------------------------------------------------------------------
      INTEGER    NF      , LI      , LF
C-----------------------------------------------------------------------
```

I4JGAM

```
      FUNCTION I4JGAM( INDEX )
      IMPLICIT NONE
C
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 INTEGER FUNCTION: I4JGAM ****************
C
C  PURPOSE:
C
C           USES INDEX TO REFERENCE 'JGAM' TABLE GENERATED BY SUBROUTINE
C           'XXGAMA'.
C
C           THE 'GAM' TABLE IS REFERENCED WITH THE FUNCTION 'R8GAM'.
C
C
C  CALLING PROGRAM: GENERAL USE.
C
C  FUNC  : (I*4)  I4JGAM  =
C
C  INPUT : (I*4)  INDEX   =
C
C  PARAM : (I*4)  MXINDX  = 200
C
C          (L*4)  LFIRST   = .TRUE.  = FIRST TIME FUNCTION CALLED.
C                            .FLASE. = FUNCTION HAS BEEN CALLED BEFORE.
C
C          (I*4)  JGAM()  =
C                           DIMENSION: REFERENCED BY 'INDEX'.
C          (I*4)  GAM()   =
C                           DIMENSION: REFERENCED BY 'INDEX'.
C
```

```
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUPUT OF MESSAGES.
C          XXGAMA     ADAS      FILLS 'JGAM'.
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     30/09/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER    I4JGAM
C-----------------------------------------------------------------------
      INTEGER    I4UNIT
C-----------------------------------------------------------------------
      INTEGER    MXINDX
      PARAMETER( MXINDX = 200 )
C-----------------------------------------------------------------------
      INTEGER    INDEX
C-----------------------------------------------------------------------
      LOGICAL    LFIRST
C-----------------------------------------------------------------------
      INTEGER    JGAM(MXINDX)
C-----------------------------------------------------------------------
      REAL*8     GAM(MXINDX)
C-----------------------------------------------------------------------
```

I4JGRP:

```
      FUNCTION I4JGRP(CHR)
C-----------------------------------------------------------------------
C***********************************************************************
C  RETURNS DECIMAL FORM OF EISSNER SINGLE HEX CHARACTER
C  ORBITAL GIVEN IN THE EISSNER SINGLE HEXADECIMAL CHARACTER FORM
C    INPUT
C           CHR= HEX. ORBITAL CHARACTER (EG. 'A')
C    OUTPUT
C           I4JGRP= DECIMAL ORBITAL CHARACTER (E.G. '10')
C
C********************** W.J.DICKSON  JET  20/10/89 *******************
C UNIX-IDL PORT:
C           WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    22ND APRIL 1996
C
C VERSION: 1.1                              DATE: 22-04-96
C MODIFIED: WILLIAM OSBORN
C            - FIRST VERSION. NO CHANGES TO IBM CODE.
C-----------------------------------------------------------------------
      CHARACTER*1 CHR,CHRA
      DIMENSION CHRA(15),JGRPA(15)
C-----------------------------------------------------------------------
```

I4LGRP

```
            FUNCTION I4LGRP(CHR)
C-----------------------------------------------------------------------
C***********************************************************************
C  RETURNS ANGULAR MOMENTUM QUANTUN NUMBER OF
C  ORBITAL GIVEN IN THE EISSNER SINGLE HEXADECIMAL CHARACTER FORM
C    INPUT
C           CHR= HEX. ORBITAL CHARACTER (EG. 'A')
C    OUTPUT
C           I4LGRP= DECIMAL ORBITAL CHARACTER (E.G. '0')
C
C********************** W.J.DICKSON  JET  20/10/89 *******************
C UNIX-IDL PORT:
C           WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    22ND APRIL 1996
C
C VERSION: 1.1                              DATE: 22-04-96
C MODIFIED: WILLIAM OSBORN
C            - FIRST VERSION. NO CHANGES TO IBM CODE.
C-----------------------------------------------------------------------
      CHARACTER*1 CHR,CHRA
      DIMENSION CHRA(15),LGRPA(15)
C-----------------------------------------------------------------------
```

I4LGRP

```
      FUNCTION I4LGRP(CHR)
C-----------------------------------------------------------------------
C***********************************************************************
```

```
C  RETURNS ANGULAR MOMENTUM QUANTUN NUMBER OF
C  ORBITAL GIVEN IN THE EISSNER SINGLE HEXADECIMAL CHARACTER FORM
C    INPUT
C           CHR= HEX. ORBITAL CHARACTER (EG. 'A')
C    OUTPUT
C           I4LGRP= DECIMAL ORBITAL CHARACTER (E.G. '0')
C
C********************** W.J.DICKSON  JET  20/10/89 *******************
C UNIX-IDL PORT:
C          WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    22ND APRIL 1996
C
C VERSION: 1.1                            DATE: 22-04-96
C MODIFIED: WILLIAM OSBORN
C           - FIRST VERSION. NO CHANGES TO IBM CODE.
C-----------------------------------------------------------------------
      CHARACTER*1 CHR,CHRA
      DIMENSION CHRA(15),LGRPA(15)
C-----------------------------------------------------------------------
```

I4NGRP

```
      FUNCTION I4NGRP(CHR)
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: I4NGRP  *******************
C
C  PURPOSE: RETURNS N QUANTUM NUMBER GIVEN IN THE EISSNER SINGLE
C           HEXADECIMAL CHARACTER FORM
C
C
C  INPUT :  (C*1)   CHR    = HEX. ORBITAL CHARACTER (EG. 'A')
C
C  OUTPUT:  (I*4)   I4NGRP = PRINCIPAL QUANTUM NUMBER FOR ORBITAL
C
C           (I*4)   I      = GENERAL INDEX
C
C  SUBROUTINES:  NONE
C  AUTHOR:  W.J. DICKSON, JET
C
C  DATE:    20/10/89
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                            DATE: 22-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C VERSION: 1.2                            DATE: 22-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT IN DEFAULT OF ZERO FOR RETURN VALUE.
C
C-----------------------------------------------------------------------
      INTEGER      I
C-----------------------------------------------------------------------
      CHARACTER*1  CHR    , CHRA(15)
C-----------------------------------------------------------------------
      INTEGER      I4NGRP , NGRPA(15)
C-----------------------------------------------------------------------
```

I4PGRP

```
      FUNCTION I4PGRP(CHR)
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: I4PGRP  *******************
C
C  PURPOSE: RETURNS PARITY OF ORBITAL GIVEN THE EISSNER SINGLE
C           HEXADECIMAL CHARACTER FORM
C
C
C  INPUT :  (C*1)   CHR    = HEX. ORBITAL CHARACTER (EG. 'A')
C
C  OUTPUT:  (I*4)   I4PGRP = PARITY FOR ORBITAL
C
C           (I*4)   I      = GENERAL INDEX
C
C  SUBROUTINES:  NONE
C  AUTHOR:  W.J. DICKSON, JET
C
C  DATE:    20/10/89
C
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                            DATE: 22-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
```

```
C VERSION: 1.2                          DATE: 22-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT IN DEFAULT OF ZERO FOR RETURN VALUE
C
C-----------------------------------------------------------------------
      INTEGER      I
C-----------------------------------------------------------------------
      CHARACTER*1  CHR    , CHRA(15)
C-----------------------------------------------------------------------
      INTEGER      I4PGRP , IPGRPA(15)
C-----------------------------------------------------------------------
```

I4SCHR

```
       FUNCTION I4SCHR(CHR,IABT)
       IMPLICIT NONE
C-----------------------------------------------------------------
C  ************* FORTRAN77 FUNCTION: I4SCHR ********************
C
C PURPOSE: CONVERT FROM CHARACTER REPRESENTATION OF NUMBER OF
C          EQUIVALENT ELECTRONS TO DECIMAL FORM
C
C  FUNCTION:
C  INPUT : (C*1)   CHR     = CHARACTER FORM FOR THE NO. OF
C                            EQUIVALENT ELECTRONS
C  OUTPUT: (I*4)   I4SCHR  = INTEGER FORM FOR THE NO. OF
C                            EQUIVALENT ELECTRONS
C          (I*4)   IABT    = RETURN CODE:
C                                0 => NO ERROR
C                                1 => ERROR (A VALUE 'I4SCHR=0' WILL BE
C                                           RETURNED).
C          (I*4)   I       = GENERAL INTEGER
C
C  SUBROUTINES: NONE
C
C  AUTHOR: H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C
C  DATE:  30/10/95
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 22-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------
       INTEGER   I4SCHR  , I      , IABT
C-----------------------------------------------------------------
       CHARACTER CHR*1
C-----------------------------------------------------------------
       CHARACTER CHRA(15)*1
C-----------------------------------------------------------------
```

I4UNIT

```
      FUNCTION I4UNIT( IUNIT )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ************** FORTRAN77 INTEGER*4 FUNCTION: I4UNIT *****************
C
C  PURPOSE: TO RESET OR RETURN A STORED INTEGER*4 VALUE GREATER THAN OR
C           EQUAL TO ZERO.
C           THIS IS USED WITHIN ADAS TO STORE THE STREAM/UNIT NUMBER
C           FOR THE OUTPUT OF ERROR MESSAGES (TO THE SCREEN).
C
C           BY DEFAULT THE STORED VALUE WILL BE 6, AND WILL BE RETURNED
C           BY THE FUNCTION IF IUNIT ON INPUT < 0.
C
C           TO RESET THE STORED VALUE THEN SET IUNIT TO THE REQUIRED
C           POSITIVE INTEGER (INC. ZERO). THIS VALUE WILL ALSO BE
C           RETURNED BY THE FUNCTION.
C
C                 IUNIT VALUE              RETURNED FUNCTION VALUE
C                 -----------              -----------------------
C                 IUNIT <  0           = CURRENT STORED INTEGER VALUE
C                                        (6 BY DEFAULT).
C                 IUNIT >= 0           = IUNIT , AND RESETS THE STORED
C                                                VALUE TO IUNIT.
C
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  O     : (I*4)  I4UNIT   = FUNCTION NAME - (SEE ABOVE)
C
C  I     : (I*4)  IUNIT    = FUNCTION ARGUMENT - (SEE ABOVE)
C
C          (I*4)  IDEFLT   = PARAMETER = DEFAULT STORED INTEGER VALUE
C
C          (I*4)  ICURNT   = CURRENT STORED INTEGER VALUE
```

```
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 5023
C
C DATE:    23/04/93
C
C UPDATE:  24/05/93 - PE BRIDEN - ALLOWED 0 TO BE A VALID STORED NUMBER
C
C-----------------------------------------------------------------------
      INTEGER    I4UNIT     , IDEFLT
C-----------------------------------------------------------------------
      PARAMETER( IDEFLT = 6 )
C-----------------------------------------------------------------------
      INTEGER    IUNIT      , ICURNT
C-----------------------------------------------------------------------
```

LENSTR

```
      FUNCTION LENSTR(ASTR)
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 FUNCTION: LENSTR **********************
C
C
C***********************************************************************C
C                                                                       C
C     FUNCTION : RETURNS THE EFFECTIVE LENGTH OF A GIVEN STRING         C
C                (IGNORING TRAILING BLANKS)                             C
C                                                                       C
C                                                                       C
C***********************************************************************C
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 18-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
C
      CHARACTER*(*) ASTR
C-----------------------------------------------------------------------
```

LINFIT

```
       SUBROUTINE LINFIT(X,XSA,Y,YSA,ICT)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 SUBROUTINE: LINFIT ********************
C
C PURPOSE: SUBROUTINE TO PERFORM LINEAR INTERPOLATION
C
C  INPUT
C      X     = REQUIRED X-VALUE
C      XSA(I)= X-VALUES
C      YSA(I)= Y-VALUES
C      ICT   = NUMBER OF VALUES
C  OUTPUT
C      Y     = RETURNED Y-VALUE
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-03-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER S.C.C.S. CONTROL
C
C-----------------------------------------------------------------------
```

MATINV

```
      SUBROUTINE MATINV(A,N,B,M,DETERM)
      IMPLICIT REAL*8 (A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 PROGRAM: MATINV ***********************
C
C MATRIX INVERSION WITH ACCOMPANYING SOLUTION OF LINEAR EQUATIONS
```

```
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 12-12-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C-----------------------------------------------------------------------
      DIMENSION IPIVOT(30),A(30,30),B(30),INDEX(30,2),PIVOT(30)
C-----------------------------------------------------------------------
```

NSORT

```
      SUBROUTINE NSORT(XA,IA,N)
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 SUBROUTINE: NSORT *********************
C
C PURPOSE: SUBROUTINE TO SORT AN ARRAY SO THAT IT IS INCREASING ORDER
C
C
C INPUT
C     XA(I)=X-VALUES
C     IA(I)=I-VALUES
C     N=NUMBER OF VALUES
C OUTPUT
C     XA(I)=SORTED X-VALUES
C     IA(I)=SORTED I-VALUES
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 01-02-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
      REAL XA(*),SWAP
      INTEGER IA(*),ISWAP
C-----------------------------------------------------------------------
```

R8AH

```
      FUNCTION R8AH ( NU , LU , NL , LL )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 REAL*8 FUNCTION: R8AH   *****************
C
C  PURPOSE: CALCULATES A-VALUES FOR HYDROGEN.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC:   (R*8)  R8AH    = HYDROGEN A-VALUE.
C
C  INPUT:  (I*4)  NU      = UPPER VALUE OF N QUANTUM NUMBER.
C  INPUT:  (I*4)  LU      = UPPER VALUE OF L QUANTUM NUMBER.
C  INPUT:  (I*4)  NL      = LOWER VALUE OF N QUANTUM NUMBER.
C  INPUT:  (I*4)  LL      = LOWER VALUE OF L QUANTUM NUMBER.
C
C  PARAM:  (R*8)  P1      = EQUATION CONSTANT.
C
C          (R*8)  XNU     = REAL VALUE = NU.
C          (R*8)  XLU     = REAL VALUE = LU.
C          (R*8)  XNL     = REAL VALUE = NL.
C          (R*8)  XLL     = REAL VALUE = LL.
C          (R*8)  T1      =
C          (R*8)  DE      =
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          R8RD2B     ADAS      RETURNS HYDRONIC BOUND-BOUND RADIAL
C                               INTEGRAL.
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     29/09/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8     R8AH
C-----------------------------------------------------------------------
      REAL*8     R8RD2B
C-----------------------------------------------------------------------
      REAL*8     P1
```

```
      PARAMETER( P1 = 2.67744D+09 )
C-----------------------------------------------------------------------
      INTEGER    NU      , LU      , NL      , LL
C-----------------------------------------------------------------------
      REAL*8     XNU     , XLU     , XNL     , XLL     ,
     &           T1      , DE
C-----------------------------------------------------------------------
```

R8ATAB

```
      FUNCTION R8ATAB ( IZ1 , NU , LU , NL , LL )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 REAL*8 FUNCTION: R8ATAB *****************
C
C  PURPOSE: CALCULATES HYDRONIC L RESOLVED A-VALUES.
C
C           ON FIRST CALL WITH VALID INPUT PARAMETERS, FUNCTION SETS UP
C           A TABLE OF A-VALUES. ON SUBSEQUENT CALLS VALUES ARE THEN
C           LOOKED UP.
C
C           FUNCTION CHECKS TO SEE IF A-VALUE IS POSSIBLE AND DIPOLE
C           ALLOWED AND RETURNS ZERO IF NOT.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC:   (R*8)  R8ATAB   = HYDROGEN A-VALUE.
C
C  INPUT:  (I*4)  IZ1      = ION CHARGE.
C  INPUT:  (I*4)  NU       = UPPER VALUE OF N QUANTUM NUMBER.
C  INPUT:  (I*4)  LU       = L QUANTUM NUMBER FOR NU.
C  INPUT:  (I*4)  NL       = LOWER VALUE OF N QUANTUM NUMBER.
C  INPUT:  (I*4)  LL       = L QUANTUM NUMBER FOR NL.
C
C  PARAM:  (I*4)  MXN      = MAXIMUM VALUE OF N QUANTUM NUMBER.
C
C          (I*4)  M        = TABLE INDEX.
C          (I*4)  N1       = N QUANTUM NUMBER LOOP INDEX.
C          (I*4)  L1       = N QUANTUM NUMBER LOOP INDEX.
C          (I*4)  N2       = N QUANTUM NUMBER LOOP INDEX.
C          (I*4)  L2       = N QUANTUM NUMBER LOOP INDEX.
C          (I*4)  J0       = TABLE INDEX.
C
C          (R*8)  Z14      = REAL VALUE = IZ1**4
C
C          (L*4)  LFIRST   = .TRUE.  = FIRST TIME FUNCTION CALLED.
C                            .FLASE. = FUNCTION HAS BEEN CALLED BEFORE.
C
C          (I*4)  MA()     = INDEX TABLE FOR 'ATABLE'.
C                            DIMENSION: REFERENCED BY NU QUANTUM NO.
C          (R*8)  ATABLE() = TABLE OF A-VALUES.
C                            DIMENSION: REFERENCED BY NU, LU, NL, LL
C                                       QUANTUM NUMBERS.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUPUT OF MESSAGES.
C          R8AH       ADAS      RETURNS HYDRONIC A-VALUE.
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     29/09/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8     R8ATAB
C-----------------------------------------------------------------------
      INTEGER    I4UNIT
      REAL*8     R8AH
C-----------------------------------------------------------------------
      INTEGER    MXN
      PARAMETER( MXN = 20 )
C-----------------------------------------------------------------------
      INTEGER    IZ1     , NU      , LU      , NL      , LL
      INTEGER    M       , N1      , L1      , N2      , L2      ,
     &           J0
C-----------------------------------------------------------------------
      REAL*8     Z14
C-----------------------------------------------------------------------
      LOGICAL    LFIRST
C-----------------------------------------------------------------------
      INTEGER    MA(MXN)
C-----------------------------------------------------------------------
      REAL*8     ATABLE((MXN+1)*(MXN+2)*(2*MXN-3)/6-(MXN+1)*(MXN-2))
C-----------------------------------------------------------------------
```

R8BCON

```
      FUNCTION R8BCON( INTYP , OUTTYP , EIN )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 REAL*8 FUNCTION: R8ECON *****************
C
C  PURPOSE: TO CONVERT A BEAM ENERGY INTO SPECIFIED UNITS
C           (DOUBLE PRECISION FUNCTION VERSION OF 'XXBCON')
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C             (R*8)  R8BCON  = FUNCTION NAME -
C                              OUTPUT BEAM ENERGY (STATED UNITS)
C             (I*4)  INTYP   = 1 => 'EIN' UNITS: EV/AMU
C                              = 2 => 'EIN' UNITS: AT. UNITS
C                              = 3 => 'EIN' UNITS: CM S-1.
C             (I*4)  OUTTYP  = 1 => 'R8BCON' UNITS: EV/AMU
C                              = 2 => 'R8BCON' UNITS: AT. UNITS
C                              = 3 => 'R8BCON' UNITS: CM S-1.
C             (R*8)  EIN     = INPUT BEAM ENERGY (STATED UNITS)
C
C             (R*8)  EVUATU  = EV/AMU  TO AT. UNITS.
C             (R*8)  EVUCMS  = EV/AMU TO CM S-1.
C
C             (R*8)  BCONV() = BEAM ENERGY CONVERSION PARAMETERS
C
C ROUTINES:  NONE
C
C NOTE:
C             BEAM ENERGY CONVERSION PARAMETERS:
C
C             INTYP = 1 ; BCONV(1) =>             EV/AMU  -> OUTPUT UNITS
C             INTYP = 2 ; BCONV(2) =>          AT. UNITS  -> OUTPUT UNITS
C             INTYP = 3 ; BCONV(3) =>             CM S-1  -> OUTPUT UNITS
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE:    21/04/95
C
C UPDATE:  15/05/95       Tim Hammond      UNIX PORT
C                         Put under SCCS control
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8     EVUATU             , EVUCMS
C-----------------------------------------------------------------------
      PARAMETER( EVUATU=6.32678D-03 , EVUCMS=1.38410D+06 )
C-----------------------------------------------------------------------
      INTEGER    INTYP              , OUTTYP
C-----------------------------------------------------------------------
      REAL*8     R8BCON             , EIN
      REAL*8     BCONV(3)
C-----------------------------------------------------------------------
```

R8DCON

```
      FUNCTION R8DCON( INTYP, OUTTYP, IZ1, DIN )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 REAL*8 FUNCTION: R8DCON *****************
C
C  PURPOSE: TO CONVERT A DENSITY INTO SPECIFIED UNITS
C           (DOUBLE PRECISION FUNCTION VERSION OF 'XXTCON')
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C             (R*8)  R8DCON  = FUNCTION NAME -
C                              OUTPUT DENSITY (STATED UNITS)
C             (I*4)  INTYP   = 1 => 'DIN' UNITS: CM-3
C                              = 2 => 'DIN' UNITS: REDUCED
C             (I*4)  OUTTYP  = 1 => 'R8DCON' UNITS: CM-3
C                              = 2 => 'R8DCON' UNITS: REDUCED
C             (I*4)  IZ1     = RECOMBINING ION CHARGE (= Z+1).
C             (R*8)  DIN     = INPUT DENSITY (STATED UNITS)
C
C             (R*8)  Z1P7   = 'IZ1'**7
C             (R*8)  DCONV() = DENSITY CONVERSION PARAMETERS
C
C ROUTINES:  NONE
C
C NOTE:
C             DENSITY CONVERSION PARAMETERS:
C
C             INTYP = 1 ; DCONV(1) =>              CM-3  -> OUTPUT UNITS
C             INTYP = 2 ; DCONV(2) =>           REDUCED  -> OUTPUT UNITS
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
```

```
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    04/01/91
C
C UPDATE:  07/08/91 - PE BRIDEN: CHANGED 'DBLE(IZ1**7)' TO 'DBLE(IZ1)**7
C                                TO AVOID INTEGER OVERFLOW IF IZ1>21.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER    INTYP              , OUTTYP            ,
     &           IZ1
C-----------------------------------------------------------------------
      REAL*8     R8DCON             , DIN
      REAL*8     Z1P7               , DCONV(2)
C-----------------------------------------------------------------------
```

R8ECIP

```
      FUNCTION R8ECIP( IZC , XI , ZETA , TE )
C
      IMPLICIT NONE
C
C-----------------------------------------------------------------------
C
C  ********** FORTRAN77 REAL*8 FUNCTION: R8ECIP *****************
C
C  PURPOSE:  CALUCLATES THE SHELL CONTRIBUTION TO THE IONISATION RATE
C            COEFFICIENT IN THE ECIP APPROXIMATION OF BURGESS.
C            CF. SUMMERS (1974) APPLETON LABORATORY REPORT IM367.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC:   (R*8)  R8ECIP  = IONISATION RATE COEFFICIENT.
C                           UNITS: CM3 SEC-1
C
C  INPUT : (I*4)  IZC     = TARGET ION CHARGE NUMBER.
C  INPUT : (R*8)  XI      = EFFECTIVE IONISATION POTENTIAL FOR SHELL.
C                           UNITS: RYD
C  INPUT : (R*8)  ZETA    = EFFECTIVE NUMBER OF EQUIVALENT ELECTRONS FOR
C                           SHELL.
C  INPUT : (R*8)  TE      = ELECTRON TEMPERATURE.
C                           UNITS: K
C
C  PARAM : (I*4) MXT      = 5.
C  PARAM : (R*8) P1       =
C
C          (I*4)  J       = LOOP INDEX
C
C          (R*8)  Z       = REAL VALUE = IZC+1.
C          (R*8)  ATE     =
C          (R*8)  EN      =
C          (R*8)  Y       =
C          (R*8)  AI      =
C          (R*8)  B       =
C          (R*8)  B1      =
C          (R*8)  C       =
C          (R*8)  R       =
C          (R*8)  D       =
C          (R*8)  F       =
C          (R*8)  C1      =
C          (R*8)  C2      =
C          (R*8)  C4      =
C
C          (R*8)  X()     =
C          (R*8)  W()     =
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          R8YIP      ADAS
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    03/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8     R8ECIP
C-----------------------------------------------------------------------
      REAL*8     R8YIP
```

```
C-----------------------------------------------------------------------
      INTEGER    MXT
      PARAMETER( MXT = 5 )
C-----------------------------------------------------------------------
      REAL*8     P1
      PARAMETER( P1 = 157890.0D0 )
C-----------------------------------------------------------------------
      INTEGER    IZC
      INTEGER    J
C-----------------------------------------------------------------------
      REAL*8     XI      , ZETA    , TE
      REAL*8     Z       , ATE     , EN      , Y       , AI      ,
     &           B       , B1      , C       , R       , D       ,
     &           F       , C1      , C2      , C4
C-----------------------------------------------------------------------
      REAL*8     X(MXT)  , W(MXT)
C-----------------------------------------------------------------------
```

R8ECON

```
      FUNCTION R8ECON( INTYP, OUTTYP, EIN )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: R8ECON *********************
C
C  PURPOSE: TO CONVERT A VELOCITY/ENERGY INTO A SPECIFIED FORM
C           (DOUBLE PRECISION FUNCTION VERSION OF 'XXECON')
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C             (R*8)  R8ECON  = FUNCTION NAME -
C                              OUTPUT VELOCITY/ENERGY (STATED UNITS)
C             (I*4)  INTYP   = 1 => 'EIN' UNITS: AT. UNITS (VEL.)
C                              = 2 => 'EIN' UNITS: CM/SEC    (VEL.)
C                              = 3 => 'EIN' UNITS: EV/AMU  (ENERGY)
C             (I*4)  OUTTYP  = 1 => 'R8ECON' UNITS: AT.UNITS (VEL.)
C                              = 2 => 'R8ECON' UNITS: CM/SEC   (VEL.)
C                              = 3 => 'R8ECON' UNITS: EV/AMU (ENERGY)
C             (R*8)  EIN     = INPUT  VELOCITY/ENERGY (STATED UNITS)
C
C             (R*8)  AMU2KG  = PARAMETER: AMU TO KG CONVERSION FACTOR
C             (R*8)  EV2J    = PARAMETER: EV TO JOULES CONVERSION FACTOR
C             (R*8)  M2CM    = PARAMETER: METRES TO CM CONVERSION FACTOR
C             (R*8)  VELE0H  = PARAMETER: ORBITAL VELOCITY  (CM/SEC)  OF
C                              AN ELECTRON IN THE SMALLEST  ORBIT  OF  A
C                              HYDROGEN ATOM (BOHR) = 2.1877D+8 CM/SEC
C             (R*8)  AT2VEL  = AT.UNITS (VEL) TO CM/SEC (VEL) CONVERSION
C             (R*8)  VEL2AT  = CM/SEC (VEL) TO AT.UNITS (VEL) CONVERSION
C             (R*8)  VEL2EN  = CM/SEC (VEL) TO EV/AMU (ENGY.) CONVERSION
C             (R*8)  EN2VEL  = EV/AMU (ENGY.) TO CM/SEC (VEL) CONVERSION
C             (R*8)  AT2EN   = AT.UNITS (VEL) TO EV/AMU (ENG) CONVERSION
C             (R*8)  EN2AT   = EV/AMU (ENG) TO AT.UNITS (VEL) CONVERSION
C
C             (R*8)  ECONV() = ENERGY/VELOCITY CONVERSION PARAMETERS
C
C ROUTINES:  NONE
C
C NOTE:
C            ENERGY/VELOCITY CONVERSION PARAMETERS:
C
C            INTYP = 1 ; ECONV(1) =>  VELOCITY: AT.UNITS -> OUTPUT FORM
C            INTYP = 2 ; ECONV(2) =>  VELOCITY: CM/SEC   -> OUTPUT FORM
C            INTYP = 3 ; ECONV(3) =>  ENERGY  : EV/AMU   -> OUTPUT FORM
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    05/02/91
C
C VERSION: 1.2                           DATE: 01-05-96
C MODIFIED: WILLIIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C            - CORRECTED AT2EN. WAS AT2VEL*VEL2EN
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8     AMU2KG                , EV2J                  ,
     &           M2CM                  , VELE0H
      REAL*8     AT2VEL                , VEL2AT
      REAL*8     VEL2EN                , EN2VEL
      REAL*8     AT2EN                 , EN2AT
C-----------------------------------------------------------------------
      PARAMETER( AMU2KG = 1.66053D-27    , EV2J   = 1.602192D-19  ,
     &           M2CM   = 1.0D+02        , VELE0H = 2.1877D+08    )
      PARAMETER( AT2VEL = VELE0H         , VEL2AT = 1.0/VELE0H    )
      PARAMETER( VEL2EN = AMU2KG/( 2.0*EV2J*M2CM*M2CM )           ,
     &           EN2VEL = ( 2.0*EV2J*M2CM*M2CM )/AMU2KG           )
      PARAMETER( AT2EN  = AT2VEL*AT2VEL*VEL2EN                    ,
     &           EN2AT  = EN2VEL*VEL2AT*VEL2AT                    )
C-----------------------------------------------------------------------
      INTEGER    INTYP                 , OUTTYP
```

```
C-----------------------------------------------------------------------
      REAL*8     R8ECON                , EIN
      REAL*8     ECONV(3)
C-----------------------------------------------------------------------
```

R8FBCH

```
      FUNCTION R8FBCH( IZ , XI , ZETA , TE )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 REAL*8 FUNCTION: R8FBCH *****************
C
C  PURPOSE: EVALUATES A SHELL CONTRIBUTION TO THE IONISATION RATE COEFF-
C           ICIENT IN THE BURGESS-CHIDICHIMO APPROXIMATION.
C
C  REFERENCE: MNRAS.(1983)203,1269.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C          (R*8)  R8FBCH  = FUNCTION NAME
C          (I*4)  IZ      = TARGET ION CHARGE NUMBER
C                           (RECOMBINED ION CHARGE).
C          (R*8)  XI      = EFFECTIVE IONISATION POTENTIAL FOR SHELL
C                           (UNITS: RYDBERGS)
C                           (LEVEL ENERGY RELATIVE TO IONISATION POT.)
C          (R*8)  ZETA    = EFFECTIVE NUMBER OF EQUIVALENT ELECTRONS
C                           IN SHELL
C          (R*8)  TE      = ELECTRON TEMPERATURE (IN KELVIN)
C
C          (R*8)  C       = PARAMETER = EQUATION CONSTANT = 2.3
C          (R*8)  DXIPOW  = PARAMETER = EQUATION CONSTANT = 1.5
C          (R*8)  TK2ATE  = PARAMETER = EQUATION CONSTANT = 1.5789D+05
C          (R*8)  R2GAM   = PARAMETER = EQUATION CONSTANT = 2.17161D-08
C          (R*8)  D150    = PARAMETER = 150
C          (R*8)  CR2GAM  = PARAMETER = EQUATION CONSTANT = 'C'*'R2GAM'
C
C          (R*8)  BETA    = EQUATION CONSTANT (SEE NOTE BELOW)
C          (R*8)  Y       =
C          (R*8)  T1      =
C          (R*8)  W       =
C          (R*8)  P       = TEMPORARY PARAMETER STORAGE
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          R8FEEI     ADAS      FUNCTION:
C
C NOTE:
C                              (100*IZ) + 91
C                      SQRT( ------------- ) - 5
C                               (4*IZ) + 3
C           BETA =  -------------------------
C                               4
C
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     04/09/90
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      REAL*8     C                 , DXIPOW             ,
     &           TK2ATE            , R2GAM
      REAL*8     D150
      REAL*8     CR2GAM
C-----------------------------------------------------------------------
      PARAMETER( C      = 2.3D0    , DXIPOW = 1.5D0      ,
     &           TK2ATE = 1.5789D5 , R2GAM  = 2.17161D-8 )
      PARAMETER( D150   = 1.50D2   )
      PARAMETER( CR2GAM = C*R2GAM  )
C-----------------------------------------------------------------------
      INTEGER    IZ
C-----------------------------------------------------------------------
      REAL*8     R8FBCH            , XI                  ,
     &           ZETA              , TE
      REAL*8     R8FEEI            , BETA                ,
     &           Y                 , T1                  ,
     &           W                 , P
C-----------------------------------------------------------------------
```

R8FCTN

```
      FUNCTION R8FCTN( STR , IABT )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 REAL*8 FUNCTION: R8FCTN *****************
C
```

```
C  FUNCTION: TO CONVERT A FLOATING POINT NUMBER STORED IN THE STRING
C            'STR' INTO A REAL*8 VARIABLE, USING INTERNAL READ.
C            INTIALLY THE PROGRAM CHECKS TO SEE IF THE NUMBER IS OF A
C            VALID FORM.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C          (R*8)   R8FCTN  = FUNCTION NAME
C          (C*(*)) STR     = STRING CONTAINING SINGLE FLOATING POINT NO.
C          (I*4)   IABT    = RETURN CODE:
C                                0 => NO ERROR
C                                2 => ERROR (A VALUE 'R8FCTN=0.0' WILL BE
C                                       RETURNED).
C                                9 => OVERFLOW ERROR (EXPONENT  > IUOFLW)
C                                     (A VALUE 'R8FCTN=0.0' RETURNED)
C                               10 => UNDERFLOW ERROR (EXPONENT <-IUOFLW)
C                                     (A VALUE 'R8FCTN=0.0' RETURNED)
C
C          (I*4)   IUOFLW   = PARAMETER = MODULUS OF MAXIMUM ALLOWED
C                                         EXPONENT = 60
C
C          (C*1)   CH0      = PARAMETER = '0'
C          (C*1)   CH9      = PARAMETER = '9'
C          (C*1)   BLANK    = PARAMETER = ' '
C          (C*1)   CPLUS    = PARAMETER = '+'
C          (C*1)   CMINUS   = PARAMETER = '-'
C          (C*1)   CPNT     = PARAMETER = '.'
C          (C*1)   CHE      = PARAMETER = 'E'
C          (C*1)   CHD      = PARAMETER = 'D'
C
C          (I*4)   ILEN     = LENGTH OF 'STR' STRING IN BYTES
C          (I*4)   M1       = STARTING BYTE IN 'STR' OF NUMBER
C                             INCLUDING SIGN
C          (I*4)   M2       = LAST BYTE IN 'STR' OF NUMBER
C          (I*4)   IE       = STARTING BYTE OF EXPONENT IN  'STR'
C                             IGNORING ANY SIGN PRESENT.
C          (I*4)   MS       = 0 => MANTISSA HAS NO SIGN
C                             1 => MANTISSA HAS A SIGN
C          (I*4)   IS       = 0 => EXPONENT HAS NO SIGN
C                             1 => EXPONENT HAS A SIGN
C          (I*4)   IPOW     = EXPONENT
C          (I*4)   ICH0     = ICHAR('0')
C          (I*4)   ICH9     = ICHAR('9')
C          (I*4)   ISTR     = ICHAR(CURRENT BYTE POSITION IN 'STR')
C          (I*4)   I        = GENERAL USE
C
C          (L*4)   LMANT    = .TRUE.  => MANTISSA BEING ANALYSED
C                             .FALSE. => EXPONENT BEING ANALYSED
C          (L*4)   LPOINT   = .TRUE.  => DECIMAL POINT FOUND IN MANTISSA
C                             .FALSE. => NO DECIMAL POINT FOUND IN MANT.
C          (L*4)   LFOUND   = .TRUE.  => ALL OF THE INPUT NUMBER BYTES
C                                        HAVE BEEN ASSESSED.
C                             .FALSE. => INPUT NUMBER BYTES STILL BEING
C                                        ASSESSED.
C
C NOTE:     AN ERROR WILL OCCUR (IABT=2) IF THERE IS MORE THAN ONE
C           NUMBER OCCURING IN THE STRING 'STR()'
C
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     26/10/90
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER    IUOFLW
C-----------------------------------------------------------------------
      CHARACTER  CH0*1   , CH9*1   , BLANK*1   , CPLUS*1   , CMINUS*1 ,
     &           CPNT*1  , CHE*1   , CHD*1
C-----------------------------------------------------------------------
      PARAMETER( IUOFLW=60 )
      PARAMETER( CH0 ='0', CH9='9', BLANK=' ', CPLUS='+', CMINUS='-' ,
     &           CPNT='.', CHE='E', CHD='D'  )
C-----------------------------------------------------------------------
      CHARACTER  STR*(*)
C-----------------------------------------------------------------------
      INTEGER    IABT
      INTEGER    M1      , M2      , MS        , IS        , IE       ,
     &           ICH0    , ICH9    , ISTR      , ILEN      , IPOW     ,
     &           I
C-----------------------------------------------------------------------
      LOGICAL    LMANT   , LPOINT  , LFOUND
C-----------------------------------------------------------------------
      REAL*8     R8FCTN
C-----------------------------------------------------------------------
```

R8FEEI

```
      FUNCTION R8FEEI ( X )
      IMPLICIT NONE
```

```
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 REAL*8 FUNCTION: R8FEEI *****************
C
C  PURPOSE: EVALUATES THE EXPONENTIAL INTEGRAL EXP(X)E1(X)
C
C
C  REFERENCE:
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C          (R*8)  R8FEEI  = FUNCTION NAME
C          (R*8)  X       = INPUT PARAMETER
C
C          (R*8)  DHALF   = PARAMETER = 0.5
C          (R*8)  CX0     = PARAMETER = EQUATION CONSTANT
C          (R*8)  CX1     = PARAMETER = EQUATION CONSTANT
C          (R*8)  CX2     = PARAMETER = EQUATION CONSTANT
C          (R*8)  CX3     = PARAMETER = EQUATION CONSTANT
C          (R*8)  CX4     = PARAMETER = EQUATION CONSTANT
C          (R*8)  CX5     = PARAMETER = EQUATION CONSTANT
C          (R*8)  CY0     = PARAMETER = EQUATION CONSTANT
C          (R*8)  CY1     = PARAMETER = EQUATION CONSTANT
C          (R*8)  CY2     = PARAMETER = EQUATION CONSTANT
C          (R*8)  CY3     = PARAMETER = EQUATION CONSTANT
C          (R*8)  CY4     = PARAMETER = EQUATION CONSTANT
C          (R*8)  C0N     = PARAMETER = EQUATION CONSTANT
C          (R*8)  C1N     = PARAMETER = EQUATION CONSTANT
C          (R*8)  C2N     = PARAMETER = EQUATION CONSTANT
C          (R*8)  C3N     = PARAMETER = EQUATION CONSTANT
C          (R*8)  C1D     = PARAMETER = EQUATION CONSTANT
C          (R*8)  C2D     = PARAMETER = EQUATION CONSTANT
C          (R*8)  C3D     = PARAMETER = EQUATION CONSTANT
C          (R*8)  C4D     = PARAMETER = EQUATION CONSTANT
C
C          (R*8)  A       =
C          (R*8)  Y       =
C          (R*8)  Z       =
C
C ROUTINES: NONE
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     18/09/90
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      REAL*8     DHALF
      REAL*8     CX0                   , CX1                   ,
     &           CX2                   , CX3                   ,
     &           CX4                   , CX5
      REAL*8     CY0                   , CY1                   ,
     &           CY2                   , CY3                   ,
     &           CY4
      REAL*8     C0N                   , C1N                   ,
     &           C2N                   , C3N
      REAL*8     C1D                   , C2D                   ,
     &           C3D                   , C4D
C-----------------------------------------------------------------------
      PARAMETER( DHALF =  0.5D0 )
      PARAMETER( CX0 =  0.57721566D0   , CX1  =  0.99999193D0  ,
     &           CX2 =  0.24991055D0   , CX3  =  0.05519968D0  ,
     &           CX4 =  0.00976004D0   , CX5  =  0.00107857D0  )
      PARAMETER( CY0 =  1.0000000D0    , CY1  =  0.9998684D0   ,
     &           CY2 =  0.4982926D0    , CY3  =  0.1595332D0   ,
     &           CY4 =  0.0293641D0    )
      PARAMETER( C0N =  0.2677737343D0 , C1N =  8.6347608925D0 ,
     &           C2N = 18.0590169730D0 , C3N =  8.5733287401D0 )
      PARAMETER( C1D =  3.9584969228D0 , C2D = 21.0996530827D0 ,
     &           C3D = 25.6329561486D0 , C4D =  9.5733223454D0 )
C-----------------------------------------------------------------------
      REAL*8   R8FEEI                  , X
      REAL*8   A                       , Y                     ,
     &         Z
C-----------------------------------------------------------------------
```

R8FORM

```
      FUNCTION R8FORM( MXNENG , MXNSHL , N      , L      ,
     &                 IESEL  , ITYPE  , NENRGY , XLCUTA ,
     &                 PL2A   , PL3A
     &               )
C
      IMPLICIT NONE
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: R8FORM **********************
C
C  PURPOSE:  CALCULATES CHARGE EXCHANGE L-RESOLVED CROSS-SECTION AS A
C            FRACTION OF THE CORRESPONDING N-RESOLVED CROSS-SECTION.
C
```

```
C  CALLING PROGRAM: GENERAL USE.
C
C  FUNC  : (R*8)  R8FORM    = L-RESOLVED CROSS-SECTION AS FRACTION OF
C                             N-REOSLVED CROSS-SECTION.
C  INPUT : (I*4)  MXNENG    = MAXIMUM NO. OF ENERGIES.
C  INPUT : (I*4)  MXNSHL    = MAXIMUM NO. OF N SHELLS.
C  INPUT : (I*4)  N         = PRINCIPAL QUANTUM NUMBER.
C  INPUT : (I*4)  L         = ORBITAL QUANTUM NUMBER.
C  INPUT : (I*4)  IESEL     = SELECTED ENERGY INDEX.
C  INPUT : (I*4)  ITYPE     = TYPE OF APPROXIMATION TO USE.
C                           = 1 = 2L+1 INCREASE WITH L AND EXPONENTIAL
C                                 CUTOFF AS L/LCUT.
C                           = 2 = 2L+1 INCREASE WITL L, SHARP CUTOFF AT
C                                 MIN(LCUT,N-1).
C                           = 3 = (2L+1)**2 INCREASE WITH L, SHARP
C                                 CUTOFF AT MIN(LCUT,N-1).
C                           = 4 = KRONECKER DELTA(L,MIN(LCUT,N-1)).
C                           = 5 = 2L+1 INCREASE WITH L AND EXP. CUTOFF
C                                 AS MAX(0,L-LCUT)/2.
C                           = 6 = 2L+1 INCREASE WITH L AND EXP. CUTOFF
C                                 AS 2*MAX(0,L-LCUT).
C                           = 7 = NEW PRIMARY FORM BASED ON SPFMAN13
C                                 FITTING PROCEDURE WITH SHARP SWITCHING
C                                 FUNCTIONS.
C                           = 8 = NEW PRIMARY FORM BASED ON SPFMAN13
C                                 FITTING PROCEDURE WITH SOFT SWITCHING
C                                 FUNCTIONS.
C  INPUT : (I*4)  NENRGY    = NUMBER OF ENERGIES IN DATASET.
C  INPUT : (R*8)  XLCUTA()  = PARAMETERS FOR CALCULATING L-RES X-SEC.
C                             DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  PL2A()    = PARAMETERS FOR CALCULATING L-RES X-SEC.
C                             DIMENSION: ENERGY INDEX
C  INPUT : (R*8)  PL3A()    = PARAMETERS FOR CALCULATING L-RES X-SEC.
C                             DIMENSION: ENERGY INDEX
C
C  PARAM : (I*4)  MXB       = 'MXBEAM'.
C  PARAM : (I*4)  MXN       = 'MXNSHL'.
C  PARAM : (I*4)  MXTYPE    = NO. OF DIFFERENT APPROXIMATIONS.
C  PARAM : (R*8)  C1        =
C
C          (I*4)  LCUT      = CUT OFF VALUE FOR ORBITAL QUANTUM NUMBER.
C          (I*4)  LC        = CUT OFF VALUE FOR ORBITAL QUANTUM NUMBER.
C          (I*4)  NN        = PRINCIPAL QUANTUM NUMBER LOOP INDEX.
C          (I*4)  LL        = ORBITAL QUANTUM NUMBER LOOP INDEX.
C          (I*4)  IE        = ENERGY LOOP INDEX.
C
C          (R*8)  SUM       =
C          (R*8)  XLC       = REAL VALUE = LC.
C          (R*8)  XL        = REAL VALUE = L.
C          (R*8)  XLL       = REAL VALUE = LL.
C          (R*8)  EF        =
C          (R*8)  XLCRIT    =
C          (R*8)  T         =
C          (R*8)  S1        =
C          (R*8)  S2        =
C          (R*8)  T1        =
C          (R*8)  T2        =
C
C          (R*8)  SUM1A()   = TABLE OF SUMS FOR 1ST APPROXIMATION.
C                             1ST DIMENSION: L CUTOFF
C                             2ND DIMENSION: N-SHELL
C          (R*8)  SUM5A()   = TABLE OF SUMS FOR 5TH APPROXIMATION.
C                             1ST DIMENSION: L CUTOFF
C                             2ND DIMENSION: N-SHELL
C          (R*8)  SUM6A()   = TABLE OF SUMS FOR 6TH APPROXIMATION.
C                             1ST DIMENSION: L CUTOFF
C                             2ND DIMENSION: N-SHELL
C          (R*8)  SUM7A()   = TABLE OF SUMS FOR 7TH APPROXIMATION.
C                             1ST DIMENSION: ENERGY INDEX
C                             2ND DIMENSION: N-SHELL
C          (R*8)  SUM8A()   = TABLE OF SUMS FOR 8TH APPROXIMATION.
C                             1ST DIMENSION: ENERGY INDEX
C                             2ND DIMENSION: N-SHELL
C
C          (L*4)  LFIRST()  = FLAGS IF FIRST CALL OF APPROXIMATION. ONLY
C                             USED BY APPROX. WHICH REQUIRE AN INTIAL
C                             SUM
C                           = .TRUE.  = FIRST CALL.
C                           = .FALSE. = NOT FIRST CALL.
C                             1ST DIMENSION: APPROX. TYPE INDEX
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURN UNIT NO. FOR OUTPUT OF MESSAGES.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    19/10/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8     R8FORM
C-----------------------------------------------------------------------
      INTEGER    I4UNIT
```

```
C-----------------------------------------------------------------------
      INTEGER    MXE          , MXN          , MXTYPE
      PARAMETER( MXE = 30     , MXN = 20     , MXTYPE = 8 )
C-----------------------------------------------------------------------
      REAL*8     C1
      PARAMETER( C1 = 1.2D0 )
C-----------------------------------------------------------------------
      INTEGER    MXNENG  , MXNSHL  , N       , L       , IESEL   ,
     &           ITYPE   , NENRGY
      INTEGER    LCUT    , LC      , NN      , LL      , IE
C-----------------------------------------------------------------------
      REAL*8     SUM     , XLC     , XL      , XLL     , EF      ,
     &           XLCRIT  , T       , S1      , S2      , T1      ,
     &           T2
C-----------------------------------------------------------------------
      REAL*8     XLCUTA(MXNENG)  , PL2A(MXNENG)    , PL3A(MXNENG)
      REAL*8     SUM1A(MXN,MXN)  , SUM5A(MXN,MXN)  , SUM6A(MXN,MXN)  ,
     &           SUM7A(MXE,MXN)  , SUM8A(MXE,MXN)
C-----------------------------------------------------------------------
      LOGICAL    LFIRST(MXTYPE)
C-----------------------------------------------------------------------
```

R8FUN1

```
      FUNCTION R8FUN1 ( Z )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 REAL*8 FUNCTION: R8FUN1 *****************
C
C  FUNCTION: R8FUN1 =  Z
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C          (R*8)  R8FUN1  = FUNCTION NAME
C          (R*8)  Z       = INPUT VALUE
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     13/08/90
C
C-----------------------------------------------------------------------
      REAL*8 R8FUN1 , Z
C-----------------------------------------------------------------------
```

R8FUN2

```
      FUNCTION R8FUN2 ( Z )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 REAL*8 FUNCTION: R8FUN2 *****************
C
C  FUNCTION: R8FUN2 =  1 / (Z + 1)
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C          (R*8)  R8FUN2  = FUNCTION NAME
C          (R*8)  Z       = INPUT VALUE
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     13/08/90
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE
C
C-----------------------------------------------------------------------
      REAL*8 R8FUN2 , Z
C-----------------------------------------------------------------------
```

R8GAM

```
      FUNCTION R8GAM( INDEX )
C
      IMPLICIT NONE
C
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 INTEGER FUNCTION: R8GAM *****************
```

```
C
C  PURPOSE: USES INDEX TO REFERENCE 'GAM' TABLE GENERATED BY SUBROUTINE
C           'XXGAMA'.
C
C           THE 'JGAM' TABLE IS REFERENCED WITH THE FUNCTION 'I4JGAM'.
C
C
C  CALLING PROGRAM: GENERAL USE.
C
C  FUNC  : (I*4)  I4GAM   =
C
C  INPUT : (I*4)  INDEX   =
C
C  PARAM : (I*4)  MXINDX  = 200
C
C          (L*4)  LFIRST   = .TRUE.  = FIRST TIME FUNCTION CALLED.
C                            .FLASE. = FUNCTION HAS BEEN CALLED BEFORE.
C
C          (I*4)  JGAM()  =
C                           DIMENSION: REFERENCED BY 'INDEX'.
C          (I*4)  GAM()   =
C                           DIMENSION: REFERENCED BY 'INDEX'.
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUPUT OF MESSAGES.
C          XXGAMA     ADAS      FILLS 'GAM'.
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     30/09/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8     R8GAM
C-----------------------------------------------------------------------
      INTEGER    I4UNIT
C-----------------------------------------------------------------------
      INTEGER    MXINDX
      PARAMETER( MXINDX = 200 )
C-----------------------------------------------------------------------
      INTEGER    INDEX
C-----------------------------------------------------------------------
      LOGICAL    LFIRST
C-----------------------------------------------------------------------
      INTEGER    JGAM(MXINDX)
C-----------------------------------------------------------------------
      REAL*8     GAM(MXINDX)
C-----------------------------------------------------------------------
```

R8GBF

```
      FUNCTION R8GBF( EN , U )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8GBF  ******************
C
C  PURPOSE: CALCULATES HYDROGENIC N-SHELL BOUND-FREE GAUNT FACTOR GII
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC  : (R*8)  R8GBF   =
C
C  INPUT : (R*8)  EN      =
C  INPUT : (R*8)  U       = N**2 / K**2 = N**2 * E / Z**2
C
C  PARAM : (R*8)  P1      = 2/3.
C  PARAM : (R*8)  P2      = 25/18.
C  PARAM : (R*8)  P3      = 4/3
C
C          (R*8)  X       =
C          (R*8)  T1      =
C          (R*8)  T2      =
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    04/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
```

```
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8     R8GBF
C-----------------------------------------------------------------------
      REAL*8     P1                    , P2                   ,
     &           P3
      PARAMETER( P1 = 2.0D0 / 3.0D0    , P2 = 25.0D0 / 18.0D0 ,
     &           P3 = 4.0D0 / 3.0D0                            )
C-----------------------------------------------------------------------
      REAL*8     EN      , U
      REAL*8     X       , T1      , T2
C-----------------------------------------------------------------------
```

R8GIIH

```
      FUNCTION R8GIIH ( VVE    , V      , N      , L      ,
     &                  L1     , LP     , ISP    , LT     ,
     &                  LT1    , IS     , IRES
     &                )
C
      IMPLICIT NONE
C
C-----------------------------------------------------------------------
C
C  ********** FORTRAN77 REAL*8 FUNCTION: R8GIIH ******************
C
C  PURPOSE:  CALCULATES BOUND-FREE G-FACTORS FOR ANGULARLY RESOLVED
C            LEVELS USES HYDROGENIC MATRIX ELEMENTS.
C
C            FOR COMPLETENESS, THE UNRESOLVED, BUNDLED N, GBF (BURGESS
C            AND SUMMERS ,1976) CAN ALSO BE OBTAINED.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC  : (R*8)  R8GIIH  = BOUND-FREE GAUNT FACTOR
C
C  INPUT : (R*8)  VVE     = V**2*E WHERE E=(FREE ELECTRON ENERGY)/Z**2.
C                           UNITS: RYD
C  INPUT : (R*8)  V       = EFFECTIVE PRINCIPAL QUANTUM NUMBER OF BOUND
C                           ELECTRON.
C  INPUT : (I*4)  N       = PRINCIPAL QUANTUM NUMBER OF BOUND ELECTRON.
C  INPUT : (I*4)  L       = ORBITAL QUANTUM NUMBER OF BOUND ELECTRON.
C  INPUT : (I*4)  L1      = ORBITAL QUANTUM NUMBER OF FREE ELECTRON.
C  INPUT : (I*4)  LP      = TOTAL ORBITAL ANGULAR MOMENTUM QUANTUM
C                           NUMBER OF PARENT STATE.
C  INPUT : (I*4)  ISP     = 2*SP+1 WHERE SP IS TOTAL SPIN OF PARENT
C                           STATE.
C  INPUT : (I*4)  LT      = TOTAL ORBITAL ANGULAR MOMENTUM QUANTUM
C                           NUMBER OF BOUND SYSTEM.
C  INPUT : (I*4)  LT1     = TOTAL ORBITAL ANGULAR MOMENTUM QUANTUM
C                           NUMBER OF FREE SYSTEM.
C  INPUT : (I*4)  IS      = 2*S+1 WHERE S IS TOTAL SPIN OF SYSTEM.
C  INPUT : (I*4)  IRES    = LEVEL OF RESOLUTION.
C                           = 1 :
C                           = 2 : ABOVE LT1 SUM.
C                           = 3 : ABOVE LT SUM.
C                           = 4 : ABOVE S SUM.
C                           = 5 : UNRESOLVED GBF.
C
C          (R*8)  XN      = REAL VALUE = N.
C          (R*8)  XL      = REAL VALUE = L.
C          (R*8)  XL1     = REAL VALUE = L1.
C          (R*8)  XLP     = REAL VALUE = LP.
C          (R*8)  XSP     = REAL VALUE = ISP.
C          (R*8)  XLT     = REAL VALUE = LT.
C          (R*8)  XLT1    = REAL VALUE = LT1.
C          (R*8)  XS      = REAL VALUE = IS.
C          (R*8)  W       =
C          (R*8)  ANG     =
C          (R*8)  CL1     =
C          (R*8)  E2      = -E WHERE E=(FREE ELECTRON ENERGY)/Z**2.
C                           UNITS: RYD
C          (R*8)  THETH   =
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          R8WIG6     ADAS
C          R8RD2F     ADAS      RETURNS HYDROGENIC BOUND-FREE RADIAL
C                               INTEGRALS.
C          R8GBF      ADAS
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    04/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
```

```
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8     R8GIIH
C-----------------------------------------------------------------------
      REAL*8     R8WIG6  , R8RD2F  , R8GBF
C-----------------------------------------------------------------------
      INTEGER    N       , L       , L1      , LP      , ISP     ,
     &           LT      , LT1     , IS      , IRES
C-----------------------------------------------------------------------
      REAL*8     VVE     , V
      REAL*8     XN      , XL      , XL1     , XLP     , XSP     ,
     &           XLT     , XLT1    , XS      , W       , ANG     ,
     &           CL1     , E2      , THETH
C-----------------------------------------------------------------------
```

R8OVLP

```
      FUNCTION R8OVLP ( MXNSHL , NMAX , N1 , L1 , N2 , L2 )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 FUNCTION: R8OVLP **********************
C
C  PURPOSE: CALCULATES HYDROGENIC CALSSICAL OVERLAP FUNCTIONS (CF. BURGESS)
C
C  CALLING PROGRAM: ADAS308.
C
C  FUNC  : (R*8)  R8OVLP  =
C
C  INPUT : (I*4)  MXNSHL  = MAXIMUM VALUE OF N QUANTUM NUMBER.
C  INPUT : (I*4)  NMAX    = N LEVEL AT WHICH TO CHANGE APPROXIMATION.
C  INPUT : (I*4)  N1      = FIRST N QUANTUM NUMBER.
C  INPUT : (I*4)  L1      = FIRST L QUANTUM NUMBER.
C  INPUT : (I*4)  N2      = SECOND N QUANTUM NUMBER.
C  INPUT : (I*4)  L2      = SECOND L QUANTUM NUMBER.
C
C  PARAM : (I*4)  MXN     = 20.
C  PARAM : (I*4)  MXTERM  = 2*MXN.
C
C          (I*4)  N1SAVE  = N1 ON LAST CALL OF FUNCTION.
C          (I*4)  L1SAVE  = N1 ON LAST CALL OF FUNCTION.
C          (I*4)  N2SAVE  = N2 ON LAST CALL OF FUNCTION.
C          (I*4)  NA      = NUMBER OF TERMS IN POWER SERIES.
C          (I*4)  IPA     = POWER OF LEADING TERM.
C          (I*4)  NB      = NUMBER OF TERMS IN POWER SERIES.
C          (I*4)  IPB     = POWER OF LEADING TERM.
C          (I*4)  NC      = NUMBER OF TERMS IN POWER SERIES.
C          (I*4)  IPC     = POWER OF LEADING TERM.
C          (I*4)  IL2     = LOOP INDEX.
C
C          (R*8)  SOVL    =
C          (R*8)  UA      = POWER OF EXPONENT IN LEADING TERM.
C          (R*8)  UB      = POWER OF EXPONENT IN LEADING TERM.
C          (R*8)  UC      = POWER OF EXPONENT IN LEADING TERM.
C          (R*8)  S       =
C
C          (R*8)  A()     = POWER SERIES COEFFICIENTS.
C          (R*8)  B()     = POWER SERIES COEFFICIENTS.
C          (R*8)  C()     = POWER SERIES COEFFICIENTS.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      RETURNS UNIT NO. FOR OUPUT OF MESSAGES.
C          R8PROV     ADAS
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     04/10/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8     R8OVLP
C-----------------------------------------------------------------------
      INTEGER    I4UNIT
      REAL*8     R8PROV
C-----------------------------------------------------------------------
      INTEGER    MXN         , MXTERM
      PARAMETER( MXN = 20    , MXTERM = 2 * MXN )
C-----------------------------------------------------------------------
      INTEGER    MXNSHL  , NMAX    , N1      , L1      , N2      ,
     &           L2      , IL2
      INTEGER    N1SAVE  , L1SAVE  , N2SAVE  , NA      , IPA     ,
     &           NB      , IPB     , NC      , IPC
```

```
C-----------------------------------------------------------------------
      REAL*8     SOVL    , UA      , UB      , UC      , S
C-----------------------------------------------------------------------
      REAL*8     A(MXTERM)  , B(MXTERM)  , C(MXTERM)
C-----------------------------------------------------------------------
```

R8P

```
      FUNCTION R8P ( N , L )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8P    ******************
C
C  PURPOSE:
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC:   (R*8)  R8P     =
C
C  INPUT:  (I*4)  N       = PRINCIPAL QUANTUM NUMBER.
C  INPUT:  (I*4)  L       = ORBITAL QUANTUM NUMBER.
C
C          (R*8)  XN      = REAL VALUE = N.
C          (R*8)  XL      = REAL VALUE = L.
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    02/11/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8    R8P
C-----------------------------------------------------------------------
      INTEGER   N       , L
C-----------------------------------------------------------------------
      REAL*8    XN      , XL
C-----------------------------------------------------------------------
```

R8PROV

```
      FUNCTION R8PROV ( N1 , L1 , N2 , L2 )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ******************* FORTRAN77 FUNCTION: R8PROV **********************
C
C  PURPOSE:
C
C  CALLING PROGRAM: GENERAL USE.
C
C  FUNC  : (R*8)  R8PROV  =
C
C  INPUT : (I*4)  N1      = FIRST N QUANTUM NUMBER.
C  INPUT : (I*4)  L1      = FIRST L QUANTUM NUMBER.
C  INPUT : (I*4)  N2      = SECOND N QUANTUM NUMBER.
C  INPUT : (I*4)  L2      = SECOND L QUANTUM NUMBER.
C
C  PARAM : (R*8)  P1      = 0.3
C
C          (I*4)  M       =
C
C          (R*8)  XN1     = REAL VALUE = N1.
C          (R*8)  XL1     = REAL VALUE = L1.
C          (R*8)  XN2     = REAL VALUE = N2.
C          (R*8)  XL2     = REAL VALUE = L2.
C          (R*8)  XM      = REAL VALUE = M.
C          (R*8)  BT      =
C          (R*8)  X1      =
C          (R*8)  X2      =
C          (R*8)  XK      =
C
C ROUTINES: NONE
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     04/10/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8     R8PROV
C-----------------------------------------------------------------------
      REAL*8     P1
      PARAMETER( P1 = 0.3D0 )
```

```
C-----------------------------------------------------------------------
      INTEGER    N1      , L1      , N2      , L2
      INTEGER    M
C-----------------------------------------------------------------------
      REAL*8     XN1     , XL1     , XN2     , XL2     , XM      ,
     &           BT      , X1      , X2      , XK
C-----------------------------------------------------------------------
```

R8QP

```
      FUNCTION R8QP ( N , L )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 REAL*8 FUNCTION: R8QP   ******************
C
C  PURPOSE:
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC:   (R*8)  R8QP    =
C
C  INPUT:  (I*4)  N       = PRINCIPAL QUANTUM NUMBER.
C  INPUT:  (I*4)  L       = ORBITAL QUANTUM NUMBER.
C
C          (R*8)  XN      = REAL VALUE = N.
C          (R*8)  XL      = REAL VALUE = L.
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    02/11/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8    R8QP
C-----------------------------------------------------------------------
      INTEGER   N       , L
C-----------------------------------------------------------------------
      REAL*8    XN      , XL
C-----------------------------------------------------------------------
```

R8RD2B

```
      FUNCTION R8RD2B ( NU , LU , NL , LL )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 REAL*8 FUNCTION: R8RD2B ******************
C
C  PURPOSE: CALCULATES HYDROGENIC BOUND-BOUND RADIAL INTEGRALS USING
C           RECURRENCE RELATIONS.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC:   (R*8)  R8RD2B  = RESULT OF INTEGRAL.
C
C  INPUT:  (I*4)  NU      = UPPER VALUE OF N QUANTUM NUMBER.
C  INPUT:  (I*4)  LU      = UPPER VALUE OF L QUANTUM NUMBER.
C  INPUT:  (I*4)  NL      = LOWER VALUE OF N QUANTUM NUMBER.
C  INPUT:  (I*4)  LL      = LOWER VALUE OF L QUANTUM NUMBER.
C
C  PARAM:  (R*8)  P1      = EQUATION CONSTANT.
C  PARAM:  (R*8)  P12     = EQUATION CONSTANT = P1**2.
C  PARAM:  (R*8)  P2      = EQUATION CONSTANT = 1/P1.
C  PARAM:  (R*8)  P22     = EQUATION CONSTANT = P2**2.
C
C          (I*4)  JS      =
C          (I*4)  I       = LOOP INDEX.
C
C          (R*8)  XNL     = REAL VALUE = NL.
C          (R*8)  XNL2    = REAL VALUE = XNL**2.
C          (R*8)  XNU     = REAL VALUE = NU.
C          (R*8)  XNU2    = REAL VALUE = XNU**2.
C          (R*8)  XNL1    = REAL VALUE = NL-1.
C          (R*8)  U       =
C          (R*8)  V       =
C          (R*8)  W       =
C          (R*8)  P       =
C          (R*8)  XI      = REAL VALUE = I.
C          (R*8)  XI1     = REAL VALUE = I+1.
C          (R*8)  XI12    = REAL VALUE = XI1**2.
C          (R*8)  T1      =
C          (R*8)  T2      =
C          (R*8)  T3      =
C
C ROUTINES: NONE
C
```

```
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     28/09/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8     R8RD2B
C-----------------------------------------------------------------------
      REAL*8     P1              , P12             ,
     &           P2              , P22
      PARAMETER( P1 = 64.0D0     , P12 = P1**2     ,
     &           P2 = 1.0D0 / P1 , P22 = P2**2       )
C-----------------------------------------------------------------------
      INTEGER    NU      , LU      , NL      , LL
      INTEGER    JS      , I
C-----------------------------------------------------------------------
      REAL*8     XNL     , XNL1    , XNL2    , XNU     , XNU2    ,
     &           U       , V       , W       , P       ,
     &           XI      , XI1     , XI12    ,
     &           T1      , T2      , T3
C-----------------------------------------------------------------------
```

R8RD2F

```
      FUNCTION R8RD2F( N , L , L1 , E2 )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8RD2F ******************
C
C  PURPOSE: CALCULATES HYDROGENIC BOUND-FREE RADIAL INTEGRALS USING
C           RECURRENCE RELATIONS.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC:   (R*8)  R8RD2F  = RESULT OF INTEGRAL.
C
C  INPUT:  (I*4)  N       = PRINCIAPL QUANTUM NUMBER OF BOUND ELECTRON.
C  INPUT:  (I*4)  L       = ORBITAL QUANTUM NUMBER OF BOUND ELECTRON.
C  INPUT:  (I*4)  L1      = ORBITAL QUANTUM NUMBER OF FREE ELECTRON.
C  INPUT:  (R*8)  E2      =
C
C  PARAM:  (R*8)  P1      = EQUATION CONSTANT = 64.
C  PARAM:  (R*8)  P12     = EQUATION CONSTANT = P1**2.
C  PARAM:  (R*8)  P2      = EQUATION CONSTANT = 1/P1.
C  PARAM:  (R*8)  P22     = EQUATION CONSTANT = P2**2.
C
C           (I*4)  JS      =
C           (I*4)  I       = LOOP INDEX.
C
C           (R*8)  XN      = REAL VALUE = N.
C           (R*8)  XN1     = REAL VALUE = N-1.
C           (R*8)  XN2     = REAL VALUE = XN**2.
C           (R*8)  E       =
C           (R*8)  U       =
C           (R*8)  V       =
C           (R*8)  P       =
C           (R*8)  XI      = REAL VALUE = I.
C           (R*8)  XI1     = REAL VALUE = I+1.
C           (R*8)  XI12    = REAL VALUE = XI1**2.
C           (R*8)  T1      =
C           (R*8)  T2      =
C           (R*8)  T3      =
C
C ROUTINES: NONE
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     04/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8     R8RD2F
C-----------------------------------------------------------------------
      REAL*8     P1              , P12               ,
     &           P2              , P22
      PARAMETER( P1 = 64.0D0     , P12 = P1**2       ,
     &           P2 = 1.0D0 / P1 , P22 = P2**2         )
C-----------------------------------------------------------------------
```

```
      INTEGER    N       , L       , L1
      INTEGER    JS      , I
C-----------------------------------------------------------------------
      REAL*8     E2
      REAL*8     XN      , XN1     , XN2     , E       , U       ,
     &           V       , P       , XI      , XI1     , XI12    ,
     &           T1      , T2      , T3
C-----------------------------------------------------------------------
```

R8SCON

```
      FUNCTION R8SCON( INTYP, OUTTYP, XSIN )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: R8SCON *********************
C
C  PURPOSE: TO CONVERT AN ARRAY OF CROSS-SECTIONS INTO A SPECIFIED FORM.
C           (DOUBLE PRECISION FUNCTION VERSION OF 'XXSCON')
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C             (R*8)  R8SCON  = FUNCTION NAME -
C                              OUTPUT CROSS-SECTION (STATED UNITS)
C             (I*4)  INTYP   = 1 => 'XSIN' UNITS: CM**2
C                              = 2 => 'XSIN' UNITS: PI*(A0**2)
C             (I*4)  OUTTYP  = 1 => 'R8SCON' UNITS: CM**2
C                              = 2 => 'R8SCON' UNITS: PI*(A0**2)
C             (R*8)  XSIN    = INPUT  CROSS-SECTION (STATED UNITS)
C
C             (R*8)  A0      = PARAMETER: BOHR RADIUS = 5.29177D-09 cm
C             (R*8)  PI      = PARAMETER: Pi = 3.1415926536
C             (R*8)  CM2A0   = PARAMETER: CM**2 TO PI*(A0**2) CONVERSION
C                              FACTOR.
C             (R*8)  A02CM   = PARAMETER: PI*(A0**2) TO CM**2 CONVERSION
C                              FACTOR.
C
C             (R*8)  SCONV() = ENERGY/VELOCITY CONVERSION PARAMETERS
C
C ROUTINES:  NONE
C
C NOTE:
C             ENERGY/VELOCITY CONVERSION PARAMETERS:
C
C             INTYP = 1 ; SCONV(1) =>  VELOCITY: CM**2     -> OUTPUT FORM
C             INTYP = 2 ; SCONV(2) =>  VELOCITY: PI*(A0**2)-> OUTPUT FORM
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    05/02/91
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8     A0                    , PI
      REAL*8     A02CM                 , CM2A0
C-----------------------------------------------------------------------
      PARAMETER( A0     = 5.29177D-09    , PI     = 3.1415926536  )
      PARAMETER( A02CM  = PI*A0*A0       , CM2A0  = 1.0/A02CM     )
C-----------------------------------------------------------------------
      INTEGER    INTYP                 , OUTTYP
C-----------------------------------------------------------------------
      REAL*8     R8SCON                , XSIN
      REAL*8     SCONV(2)
C-----------------------------------------------------------------------
```

R8TCON

```
      FUNCTION R8TCON( INTYP , OUTTYP , IZ1 , TIN )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 REAL*8 FUNCTION: R8TCON *****************
C
C  PURPOSE: TO CONVERT A TEMPERATURE INTO SPECIFIED UNITS
C           (DOUBLE PRECISION FUNCTION VERSION OF 'XXTCON')
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNCTION:
C
C             (R*8)  R8TCON  = FUNCTION NAME -
C                              OUTPUT TEMPERATURE (STATED UNITS)
C             (I*4)  INTYP   = 1 => 'TIN' UNITS: KELVIN
C                              = 2 => 'TIN' UNITS: eV
C                              = 3 => 'TIN' UNITS: REDUCED TEMP.
C             (I*4)  OUTTYP  = 1 => 'R8TCON' UNITS: KELVIN
C                              = 2 => 'R8TCON' UNITS: eV
```

```
C                          = 3 => 'R8TCON' UNITS: REDUCED TEMP.
C             (I*4)  IZ1     = RECOMBINING ION CHARGE (= Z+1).
C             (R*8)  TIN     = INPUT TEMPERATURE (STATED UNITS)
C
C             (R*8)  EV2KEL  = ELECTRON VOLTS TO KELVIN CONVERSION
C             (R*8)  KEL2EV  = KELVIN TO ELECTRON VOLTS CONVERSION
C
C             (R*8)  Z1P2    = 'IZ1'**2
C             (R*8)  TCONV() = TEMPERATURE CONVERSION PARAMETERS
C
C ROUTINES:  NONE
C
C NOTE:
C             TEMPERATURE CONVERSION PARAMETERS:
C
C             INTYP = 1 ; TCONV(1) =>             KELVIN  -> OUTPUT UNITS
C             INTYP = 2 ; TCONV(2) =>                 eV  -> OUTPUT UNITS
C             INTYP = 3 ; TCONV(3) => REDUCED TEMPERATURE -> OUTPUT UNITS
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    04/01/91
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8     EV2KEL              , KEL2EV
C-----------------------------------------------------------------------
      PARAMETER( EV2KEL=1.16054D+04  , KEL2EV=8.61668D-05 )
C-----------------------------------------------------------------------
      INTEGER    INTYP               , OUTTYP
      INTEGER    IZ1
C-----------------------------------------------------------------------
      REAL*8     R8TCON              , TIN
      REAL*8     Z1P2                , TCONV(3)
C-----------------------------------------------------------------------
```

R8WIG6

```
      FUNCTION R8WIG6( A1 , A2 , A3 , B1 , B2 , B3 )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  *************** FORTRAN77 REAL*8 FUNCTION: R8WIG6 *****************
C
C  PURPOSE:  CALCULATES WIGNER 6J COEFFICIENTS.
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC:   (R*8)  R8WIG6  =
C
C  INPUT:  (R*8)  A1      =
C  INPUT:  (R*8)  A2      =
C  INPUT:  (R*8)  A3      =
C  INPUT:  (R*8)  B1      =
C  INPUT:  (R*8)  B2      =
C  INPUT:  (R*8)  B3      =
C
C          (I*4)  J1      =
C          (I*4)  J2      =
C          (I*4)  J3      =
C          (I*4)  J4      =
C          (I*4)  J       =
C          (I*4)  NMIN    =
C          (I*4)  NMAX    =
C          (I*4)  N1      =
C          (I*4)  N2      =
C          (I*4)  N3      =
C          (I*4)  N4      =
C          (I*4)  N5      =
C          (I*4)  N6      =
C          (I*4)  N7      =
C          (I*4)  N       = LOOP INDEX.
C          (I*4)  M1      =
C          (I*4)  I       = LOOP INDEX.
C
C          (R*8)  D1      =
C          (R*8)  D2      =
C          (R*8)  D3      =
C          (R*8)  D4      =
C          (R*8)  D       =
C          (R*8)  DSUM    =
C          (R*8)  DT      =
C          (R*8)  DP1     =
C          (R*8)  DP2     =
C          (R*8)  DP3     =
C          (R*8)  DP4     =
C          (R*8)  DP5     =
C          (R*8)  DP6     =
C          (R*8)  DP7     =
C          (R*8)  DP8     =
C
C          (I*4)  K()     =
```

```
C                                 DIMENSION: 4
C          (I*4)  L()     =
C                                 DIMENSION: 3
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4JGAM     ADAS      RETURNS VALUE FROM 'JGAM' TABLE.
C          R8GAM      ADAS      RETURNS VALUE FROM 'GAM' TABLE.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    08/10/93
C
C-----------------------------------------------------------------------
      FUNCTION R8XIP( XI , DELTA )
C
      IMPLICIT NONE
C
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8XIP  ******************
C
C  PURPOSE:
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC:   (R*8)  R8XIP   =
C
C  INPUT:  (R*8)  XI      =
C  INPUT:  (R*8)  DELTA   =
C
C          (R*8)  XM      =
C          (R*8)  X1      =
C          (R*8)  X2      =
C          (R*8)  X3      =
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    07/10/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8    R8XIP
C-----------------------------------------------------------------------
      REAL*8    XI      , DELTA
      REAL*8    XM      , X1      , X2      , X3
C-----------------------------------------------------------------------
```

R8YIP

```
      FUNCTION R8YIP( XI , DELTA )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8YIP  ******************
C
C  PURPOSE: CALCULATES BURGESS YIP FUNCTION FOR ECIP APPROXIMATION
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC:   (R*8)  R8YIP   =
C
C  INPUT:  (R*8)  XI      =
C  INPUT:  (R*8)  DELTA   =
C
C  PARAM:  (R*8)  P1      =
C
C          (I*4)  J       =
C
C          (R*8)  W       =
C          (R*8)  W1      =
C          (R*8)  W2      =
C          (R*8)  W3      =
C          (R*8)  XM      =
C          (R*8)  X       =
C          (R*8)  X0      =
C          (R*8)  X2      =
C          (R*8)  R       =
C          (R*8)  S       =
C          (R*8)  T       =
C          (R*8)  Y       =
C
C          (R*8)  A()     =
C          (R*8)  B()     =
C
C ROUTINES: NONE
C
```

```
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    07/10/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8     R8YIP
C-----------------------------------------------------------------------
      REAL*8     P1
      PARAMETER( P1 = 5.0D0 / 30.0D0 )
C-----------------------------------------------------------------------
      INTEGER    J
C-----------------------------------------------------------------------
      REAL*8     XI      , DELTA
      REAL*8     W       , W1      , W2      , W3      , XM      ,
     &           X       , X0      , X2      , R       , S       ,
     &           T       , Y
C-----------------------------------------------------------------------
      REAL*8     A(20)   , B(20)
C-----------------------------------------------------------------------
```

R8ZETA

```
      FUNCTION R8ZETA ( ZEFF , N , L )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 REAL*8 FUNCTION: R8ZETA ******************
C
C  PURPOSE: CALCULATES ZETA FUNCTION FOR BOUND-FREE QUANTUM DEFECT THEORY
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC:   (R*8)  R8ZETA  =
C
C  INPUT:  (I*4)  ZEFF    = EFFECTIVE ION CHARGE.
C  INPUT:  (I*4)  N       = PRINCIPAL QUANTUM NUMBER.
C  INPUT:  (I*4)  L       = ORBITAL QUANTUM NUMBER.
C
C          (R*8)  XN      = REAL VALUE = N.
C          (R*8)  XL      = REAL VALUE = L.
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    11/10/93
C
C-----------------------------------------------------------------------
      REAL*8    R8ZETA
C-----------------------------------------------------------------------
      INTEGER   N       , L
C-----------------------------------------------------------------------
      REAL*8    ZEFF
      REAL*8    XN      , XL
C-----------------------------------------------------------------------
```

SOLVE

```
       SUBROUTINE SOLVE(FXC2,FXC3,S,EIJ,IXTYP,TE1,GAM1,TE2,GAM2,IFAIL)
       IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: SOLVE **********************
C
C PURPOSE:  SOLVES FOR ADAS FIT PARAMETERS OF DIPOLE UPSILON AT TWO TEMPERATURES
C
C NOTES: THIS ROUTINE IS NOT YET PROPERLY ANNOTATED
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-03-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER S.C.C.S. CONTROL
C
C VERSION: 1.2                          DATE: 20-03-96
C MODIFIED: TIM HAMMOND
C               - ALTERED PRINTOUT FROM UNIT 6 TO I4UNIT(-1)
C
C-----------------------------------------------------------------------
```

XFELEM

```
      FUNCTION XFELEM ( IZ0 )
```

```
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C ************* FORTRAN77 CHARACTER*12 FUNCTION: XFELEM ****************
C
C PURPOSE: TO RETURN THE NAME OF THE ELEMENT WITH NUCLEAR CHARGE IZ0
C          (CHARACTER*12 FUNCTION VERSION OF 'XXELEM')
C
C CALLING PROGRAM: GENERAL USE
C
C FUNCTION:
C
C          (C*12) XFELEM  = FUNCTION NAME -
C                           NAME OF ELEMENT WITH NUCLEAR CHARGE 'IZ0'
C          (I*4)  IZ0     = ELEMENT NUCLEAR CHARGE
C
C          (C*12) NAMES() = NAMES OF FIRST 50 ELEMENTS.
C                           ARRAY DIMENSION => NUCLEAR CHARGE
C
C NOTES:    IF NUCLEAR CHARGE IS OUT OF RANGE, I.E.NOT BETWEEN 1 & 50,
C           THEN THE CHARACTER STRING 'XFELEM' IS RETURNED BLANK.
C
C ROUTINES: NONE
C
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     13/02/91
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER      IZ0
C-----------------------------------------------------------------------
      CHARACTER*12 XFELEM , NAMES(50)
C-----------------------------------------------------------------------
```

XFESYM

```
      FUNCTION XFESYM ( IZ0 )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C ************** FORTRAN77 CHARACTER*2 FUNCTION: XFESYM ****************
C
C PURPOSE: TO RETURN THE SYMBOL FOR THE ELEMENT WITH NUCLEAR CHARGE IZ0
C          (CHARACTER*2 FUNCTION VERSION OF 'XXESYM')
C
C CALLING PROGRAM: GENERAL USE
C
C FUNCTION:
C
C          (C*2)  XFESYM  = FUNCTION NAME -
C                           SYMBOL OF ELEMENT WITH NUCLEAR CHARGE 'IZ0'
C          (I*4)  IZ0     = ELEMENT NUCLEAR CHARGE
C
C          (C*2)  SYMBOL()= SYMBOLS OF FIRST 50 ELEMENTS.
C                           ARRAY DIMENSION => NUCLEAR CHARGE
C
C NOTES:    IF NUCLEAR CHARGE IS OUT OF RANGE, I.E.NOT BETWEEN 1 & 50,
C           THEN THE CHARACTER STRING 'XFESYM' IS RETURNED BLANK.
C
C ROUTINES: NONE
C
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     12/02/91
C
C UPDATES:  25/10/94 L. JALOTA (TESSELLA SUPPORT SERVICES PLC)
C                    CHANGED CASE OF SYMBOL TO LOWER CASE FOR UNIX
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER      IZ0
C-----------------------------------------------------------------------
      CHARACTER*2  XFESYM , SYMBOL(50)
C-----------------------------------------------------------------------
```

XX0000

```
      SUBROUTINE XX0000
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 SUBROUTINE: XX0000 *********************
C
C  PURPOSE: ADAS CONFIGURATION FILE FOR SETTING MACHINE DEPENDANT
C           VARIABLES ETC.
```

```
C
C  CALLING PROGRAM: ANY MAIN ADAS PROGRAM
C
C
C  SUBROUTINE:
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  IDUMP   = GENERAL USE
C
C NOTE:
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      SET UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 5023
C
C DATE:    23/04/93
C
C MODIFIED:  6/3/94       L. JALOTA        CHANGED OUTPUT UNIT TO 0 FOR
C                                          ERROR MESSAGES SO AS NOT TO
C                                          INTERFERE WITH POIPE COMMUN-
C                                          ICATIONS WITH IDL.
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER    I4UNIT
      INTEGER    IDUMP
C-----------------------------------------------------------------------
```

XXADAS

```
      SUBROUTINE XXADAS( HEADER )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 SUBROUTINE: XXADAS *********************
C
C  PURPOSE: ADAS ROUTINE   -  GETS AN 80 BYTE CHARACTER STRING HEADER
C           CONTAINING THE ADAS RELEASE & VERSION, THE EXECUTING PROGRAM
C           NAME & VERSION, AND THE CURRENT DATE & TIME FROM IDL VIA
C           THE PIPE.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  OUTPUT: (C*80) HEADER   = ADAS HEADER (AS DESCRIBED ABOVE)
C
C          (I*4)   PIPEIN   = PARAMETER = UNIT NUMBER FOR OUTPUT TO PIPE
C          (I*4)   PIPEOU   = PARAMETER = UNIT NUMBER FOR INPUT TO PIPE
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C AUTHOR:  ANDREW BOWEN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    24/05/93
C
C-----------------------------------------------------------------------
      CHARACTER  HEADER*80
C-----------------------------------------------------------------------
      INTEGER    PIPEIN    , PIPEOU
      PARAMETER( PIPEIN=5  , PIPEOU=6)
C-----------------------------------------------------------------------
```

XXBCON

```
      SUBROUTINE XXBCON( INTYP, OUTTYP, IEVAL, EIN, EOUT )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 SUBROUTINE: XXBCON *********************
C
C  PURPOSE: TO CONVERT AN ARRAY OF BEAM ENERGIES INTO SPECIFIED UNITS
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT :    (I*4)  INTYP   = 1 => 'EIN(ARRAY)' UNITS: EV/AMU
C                              = 2 => 'EIN(ARRAY)' UNITS: AT. UNITS
C                              = 3 => 'EIN(ARRAY)' UNITS: CM SEC-1
C  INPUT :    (I*4)  OUTTYP  = 1 => 'EOUT(ARRAY)' UNITS: EV/AMU
C                              = 2 => 'EOUT(ARRAY)' UNITS: AT. UNITS
C                              = 3 => 'EOUT(ARRAY)' UNITS: CM SEC-1
C  INPUT :    (I*4)  IEVAL   = NUMBER OF BEAM ENERGIES IN 'EIN(ARRAY)'
```

```
C  INPUT :    (R*8)  EIN()   = INPUT BEAM ENERGIES (STATED UNITS)
C  OUTPUT:    (R*8)  EOUT()  = OUTPUT BEAM ENERGIES (STATED UNITS)
C
C             (R*8)  EVUATU  = EV/AMU TO ATOMIC UNITS CONVERSION
C             (R*8)  EVUCMS  = EV/AMU TO CM S-1 CONVERSION
C
C             (I*4)  I       = GENERAL USE
C
C             (R*8)  BCONV() = BEAM ENERGY CONVERSION PARAMETERS
C
C ROUTINES:  NONE
C
C NOTE:
C            BEAM ENERGY CONVERSION PARAMETERS:
C
C            INTYP = 1 ; BCONV(1) =>            EV/AMU  -> OUTPUT UNITS
C            INTYP = 2 ; BCONV(2) =>          AT. UNITS -> OUTPUT UNITS
C            INTYP = 3 ; BCONV(3) =>          CM SEC-1  -> OUTPUT UNITS
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE:    20/04/95
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8     EVUATU             , EVUCMS
C-----------------------------------------------------------------------
      PARAMETER( EVUATU=6.32678D-03 , EVUCMS=1.38410D+06 )
C-----------------------------------------------------------------------
      INTEGER    INTYP              , OUTTYP             ,
     &           IEVAL
      INTEGER    I
C-----------------------------------------------------------------------
      REAL*8     EIN(IEVAL)         , EOUT(IEVAL)
      REAL*8     BCONV(3)
C-----------------------------------------------------------------------
```

XXCEIA

```
      SUBROUTINE XXCEIA( EIA )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXCEIA *********************
C
C  PURPOSE: CONVERTS IONISATION POTENTIALS FROM WAVE NUMBERS TO
C           RYDBERGS AND FILLS IN ANY MISSING VALUES UP TO AN ION CHARGE
C           OF 50. (IF 'EIA()' IS ALL ZERO - RETURN)
C           (PRIOR TO 16/08/90 WAS KNOWN AS 'E3EIA' - SLIGHTLY AMENDED)
C
C  CALLING PROGRAMS: GENERAL USE
C
C  SUBROUTINE:
C
C  I/O   : (R*8)  EIA()   = IONISATION POTENTIALS: ()=ION CHARGE
C  INPUT :                  UNITS - WAVE NUMBERS (CM-1)
C  OUTPUT:                  UNITS - RYDBERGS
C
C          (R*8) WN2RYD   = WAVE NUMBER (CM-1) TO RYDBERG CONVERSION
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  IPOT    = NUMBER OF IONISATION POTENTIAL VALUES
C                           PRESENT IN THE INPUT FILE.
C          (I*4)  I       = GENERAL USE
C
C          (R*8)  A1      = EXTRAPOLATION EQUATION COEFFICIENT
C          (R*8)  A2      = EXTRAPOLATION EQUATION COEFFICIENT
C          (R*8)  A3      = EXTRAPOLATION EQUATION COEFFICIENT
C          (R*8)  XI      = VALUE AT WHICH EXTRAPOLATION IS REQUIRED
C
C ROUTINES:
C          ROUTINE    SOURCE   BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS     INTEGER*4 FUNCTION -
C                              FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C
C NOTE:
C           THE NUMBER OF IONISATION COEFFICIENTS PRESENT WILL BE EITHER
C           30 OR 50. IF 30 THEN THE VALUES FROM  31  TO  50 NEED TO  BE
C           EXTRAPOLATED. THE EXTRAPOLATION EQUATION  IS  BASED  ON  THE
C           VALUES OF EIA(20), EIA(25) AND EIA(50).
C
C           THE EXTRAPOLATION EQUATION BEING:
C
C           EIA(XI) = A1 + A2*XI + A2*XI*XI
C
C ROUTINES: NONE
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
```

```
C DATE:    16/08/90
C
C UPDATE:  11/08/93  HP SUMMERS - DO NOT STOP IF IPOT.LE.30, BUT DO NOT
C                                 EXTRAPOLATE
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE (NO CHANGES)
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER   IPOT    , I      , I4UNIT
C-----------------------------------------------------------------------
      REAL*8    WN2RYD
      REAL*8    A1      , A2     , A3     , XI
      REAL*8    EIA(50)
C-----------------------------------------------------------------------
      PARAMETER (WN2RYD=9.11269D-06)
C-----------------------------------------------------------------------
```

XXCFTR

```
      SUBROUTINE XXCFTR( ICFSEL , CSTRGI , CSTRGO )
IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXCFTR *********************
C
C  PURPOSE: CONVERTS A CONFIGURATION CHARACTER STRING, SUCH AS OCCURS
C           IN A SPECIFIC ION FILE LEVEL LIST, BETWEEN EISSNER AND
C           STANDARD FORMS
C
C  CALLING PROGRAMS: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (I*4)   ICFSEL  = 1 => STANDARD FORM OUT, STANDARD FORM IN
C                            2 => EISSNER  FORM OUT, STANDARD FORM IN
C                            3 => STANDARD FORM OUT, EISSNER  FORM IN
C                            4 => EISSNER  FORM OUT, EISSNER  FORM IN
C  INPUT : (C*(*)) CSTRGI  = CONFIGURATION STRING IN INPUT FORM
C  OUTPUT: (C*(*)) CSTRGO  = CONFIGURATION STRING IN OUTPUT FORM
C
C          (I*4)   I       = GENERAL USE
C          (I*4)   ISHEL   = SHELL COUNTER
C          (I*4)   IP      = PARITY OF CONFIGURATION
C          (I*4)   MAXN    = N_SHELL SUM FOR CONFIGURATION
C          (I*4)   NSHEL   = NUMBER OF SHELLS IDENTIFIED fFROM STRING
C          (I*4)   NELA()  = NUMBER OF ELECTRONS IN EACH SHELL
C
C          (C*19)  STRG    = STANDARD FORM CONFIGURATION STRING
C          (C*19)  STRGE   = EISSNER FORM CONFIGURATION STRING
C          (C*1)   CHEISA()= EISSNER CHARACTER FOR ORBITALS
C          (C*2)   CHSTDA()= STANDARD ORBITAL SPEC. FOR EACH SHELL
C                            (EISSNER FORM CASE)
C          (C*1)   CHQA()  = INDEX TO HEXADECIMAL CONVERSIONS
C          (C*1)   CHRA()  = CHAR. FOR NO. OF. EQUIV. ELEC. IN SHELL
C                            (STANDARD FORM CASE)
C
C          (L*4)   LEISS   = .TRUE.  => EISSNER FORM
C                            .FALSE. => NOT EISSNER FORM
C
C
C ROUTINES:
C          ROUTINE    SOURCE   BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS     FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          I4NGRP     ADAS     RETURNS N QUANTUM NUMBER IN THE
C                              EISSNER SINGLE HEXADECIMAL CHARACTER FORM
C          I4PGRP     ADAS     RETURNS PARITY OF ORBITAL GIVEN THE
C                              EISSNER SINGLE HEXADECIMAL CHARACTER FORM
C          I4SCHR     ADAS     RETURNS NUMERICAL VALUE FOR NUMBER OF
C                              EQUIVALENT ELECTRONS GIVEN AS HEX> CHAR.
C          CSTGRP     ADAS     RETURNS TERM OF ORBITAL GIVEN IN THE
C                              EISSNER SINGLE HEXADECIMAL CHARACTER FORM
C          CEIGRP     ADAS     RETURNS EISSNER CODE FOR ORBITAL
C
C
C NOTE:    THE ROUTINE IS USED TO CONVERT THE CONFIGURATION CHARACTER
C          STRING OCCURRING IN ADF04 FILE LEVEL LISTS.  THE STRING
C          LENGTH ALLOCATED TO THIS IS *18 FOLLOWING 1 BLANK SPACE
C          AFTER THE LEVEL INDEX.  A PROBLEM ARISES WHEN THE FIRST
C          SHELL CONTAINS MORE THAN 9 EQUIVALENT ELECTRONS.  IN THIS
C          CASE, OVERSPILL IS ALLOWED INTO THE BLANK CHARACTER SPACE.
C          THE ROUTINE WILL ANALYSE A *19 STRING INCLUDING THE USUALLY
C          BLANK LOCATION OR A *18 STRING EXCLUDING IT.  IN THE LATTER
C          CASE AN INTELLIGENT GUESS IS MADE AS TO WHETHER THE OMITTED
C          BLANK SHOULD IN FACT BE A '1'.  THIS SITUATION OCCURS FOR A
C          LEADING CLOSED D-SHELL.
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
```

```
C DATE:    25/10/95
C
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 19-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER   ICFSEL    , I
      INTEGER   I4UNIT    , I4NGRP    , I4PGRP  , I4SCHR
      INTEGER   ISHEL     , NSHEL     , IP      , MAXN
      INTEGER   IABT
C-----------------------------------------------------------------------
      CHARACTER CSTRGI*(*), CSTRGO*(*) , CSTR19*19
      CHARACTER STRG*19   , STRGE*19
      CHARACTER CSTGRP*2  , CEIGRP*1
C-----------------------------------------------------------------------
      LOGICAL*4 LEISS
C-----------------------------------------------------------------------
      INTEGER   NELA(6)
C-----------------------------------------------------------------------
      CHARACTER CHEISA(6)*1 , CHSTDA(4)*2
      CHARACTER CHQA(15)*1  , CHRA(4)*1
C-----------------------------------------------------------------------
```

XXCHEB

```
      SUBROUTINE XXCHEB( X , Y , N , A , M1 , REF )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXCHEB *********************
C
C  PURPOSE:           CARRY OUT CHEBYSCHEV POLYNOMIAL FIT ALGORITHM
C                     (DIRECT REPLACEMENT FOR NAG MINIMAX POLYNIMIAL
C                      COEFT. ROUTINE E02ACF - has same argument list).
C
C  REFERENCE:         TRANSLATION FROM ALGOL CHEBYSCHEV POLYNOMIAL FIT
C                     ALGORITHM BY -
C                     Boothroyd,
C                     Communications of the ACM, 10(12), December 1967
C
C  CALLING PROGRAMS:  XXMNMX
C
C  SUBROUTINE:
C
C  INPUT : (R*8)  X()     = Array of Input X Co-ordinates
C                           Dimension = N
C  INPUT : (R*8)  Y()     = Array of Input Y Co-ordinates
C                           Dimension = N
C  INPUT : (I*4)  N       = Number of Data Points
C                           Dimension = N
C  OUTPUT: (R*8)  A()     = Coefficients of the Fitted Polynomial.
C                           Dimension = M1
C  INPUT : (I*4)  M1      = M + 1 = The order of the polynomial to be
C                           found + 1. The highest order term is
C                           A(M1)*X(M) !!!
C  OUTPUT: (R*8)  REF     = Final Reference Deviation.
C
C  NOTES:     Based on Revision 1.2 (13:26:01 13OCT94) of XXCHEB by CJW
C             from the University of Strathclyde with the following
C             bug correction -
C               The line (within the DO 100 loop):
C                  IF (I.NE.J)
C               was corrected to:
C                  IF (I.NE.RJ)
C
C  ROUTINES:  None
C
C  AUTHOR:    CJW (University of Strahclyde)
C  EDITED BY: PAUL BRIDEN (Tessella Support Services plc)
C             K1/0/37
C             JET ext. 5023
C
C  DATE:      31/10/94
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER    MAXNUM
C-----------------------------------------------------------------------
      PARAMETER( MAXNUM = 100 )
C
C-----------------------------------------------------------------------
      INTEGER  N            , M1
      INTEGER  M            , I           , J           ,
     &         K            , RI          , I1          ,
     &         IMAX         , RJ          , J1
C-----------------------------------------------------------------------
      REAL*8   REF
      REAL*8   D            , H           , AI1         ,
     &         RHI1         , DENOM       , AI          ,
```

```
     &         RHI          , XJ           , HMAX          ,
     &         HIMAX        , XI           , HI            ,
     &         ABSHI        , NEXTHI       , PREVH
C-----------------------------------------------------------------------
      INTEGER  R(0:MAXNUM)
C-----------------------------------------------------------------------
      REAL*8   X(N)         , Y(N)         , A(M1)
      REAL*8   RX(0:MAXNUM) , RH(0:MAXNUM) , AL(0:MAXNUM)
C
C-----------------------------------------------------------------------
```

XXDATE

```
      SUBROUTINE XXDATE ( DATE )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXDATE *********************
C
C  PURPOSE: GATHERS CURRENT DATE AS A 8 BYTE STRING FROM IDL VIA PIPE
C
C  CALLING PROGRAM: GENERAL USE
C
C
C  SUBROUTINE:
C
C
C OUTPUT:  (C*8)  DATE     = CURRENT DATE (AS 'DD/MM/YY')
C
C          (I*4)   PIPEIN   = PARAMETER = UNIT NUMBER FOR OUTPUT TO PIPE
C          (I*4)   PIPEOU   = PARAMETER = UNIT NUMBER FOR INPUT TO PIPE
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C AUTHOR:  ANDREW BOWEN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    24/05/93
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER DATE*8
C-----------------------------------------------------------------------
      INTEGER    PIPEIN    , PIPEOU
      PARAMETER( PIPEIN=5  , PIPEOU=6)
C-----------------------------------------------------------------------
```

XXDCON

```
      SUBROUTINE XXDCON( INTYP, OUTTYP, IZ1, IDVAL, DIN, DOUT )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXDCON *********************
C
C  PURPOSE: TO CONVERT AN ARRAY OF DENSITIES INTO SPECIFIED UNITS
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C  INPUT :    (I*4)  INTYP   = 1 => 'DIN(array)' UNITS: CM-3
C                              = 2 => 'DIN(array)' UNITS: REDUCED
C  INPUT :    (I*4)  OUTTYP  = 1 => 'DOUT(array)' UNITS: CM-3
C                              = 2 => 'DOUT(array)' UNITS: REDUCED
C  INPUT :    (I*4)  IZ1     = RECOMBINING ION CHARGE (= Z+1).
C  INPUT :    (I*4)  IDVAL   = NUMBER OF DENSITIES IN 'DIN(array)'
C  INPUT :    (R*8)  DIN()   = INPUT DENSITIES (STATED UNITS)
C  OUTPUT:    (R*8)  DOUT()  = OUTPUT DENSITIES (STATED UNITS)
C
C             (I*4)  I       = GENERAL USE
C
C             (R*8)  Z1P7    = 'IZ1'**7
C             (R*8)  DCONV() = DENSITY CONVERSION PARAMETERS
C
C ROUTINES:  NONE
C
C NOTE:
C            DENSITY CONVERSION PARAMETERS:
C
C            INTYP = 1 ; DCONV(1) =               CM-3  -> OUTPUT UNITS
C            INTYP = 2 ; DCONV(2) =            REDUCED  -> OUTPUT UNITS
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    04/02/91
C
C UPDATE:  07/08/91 - PE BRIDEN: CHANGED 'DBLE(IZ1**7)' TO 'DBLE(IZ1)**7
C                                TO AVOID INTEGER OVERFLOW IF IZ1>21.
C
```

```
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER    INTYP              , OUTTYP             ,
     &           IZ1                , IDVAL
      INTEGER    I
C-----------------------------------------------------------------------
      REAL*8     DIN(IDVAL)         , DOUT(IDVAL)
      REAL*8     Z1P7               , DCONV(2)
C-----------------------------------------------------------------------
```

XXDELT

```
      SUBROUTINE XXDELT( A , B , C , D , J )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXDELT ********************
C
C  PURPOSE:  EVALUATES DELTA FUNCTIONALS FOR WIGNER COEFFICIENTS
C
C  CALLING PROGRAM: GENERAL USE.
C
C  INPUT : (R*8)  A       =
C  INPUT : (R*8)  B       =
C  INPUT : (R*8)  C       =
C
C  OUTPUT: (R*8)  D       =
C  OUTPUT: (I*4)  J       =
C
C          (I*4)  J1      =
C          (I*4)  J2      =
C          (I*4)  J3      =
C          (I*4)  J4      =
C          (I*4)  J5      =
C
C          (R*8)  A1      =
C          (R*8)  A2      =
C          (R*8)  A3      =
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4JGAM     ADAS      RETURNS VALUE FROM 'JGAM' TABLE.
C          R8GAM      ADAS      RETURNS VALUE FROM 'GAM' TABLE.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    07/10/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER  I4JGAM
      REAL*8   R8GAM
C-----------------------------------------------------------------------
      INTEGER  J
      INTEGER  J1      , J2      , J3      , J4      , J5
C-----------------------------------------------------------------------
      REAL*8   A       , B       , C       , D
      REAL*8   A1      , A2      , A3
C-----------------------------------------------------------------------
```

XXDER1)

```
      SUBROUTINE XXDER1(FCN,M,N,X,FVEC,FJAC,LDFJAC,INFO,IPVT,WA,
     &                  LWA)
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C ROUTINE: XXDER1 (MINPACK ROUTINE LMDER1)
C
C PURPOSE:
C     THE PURPOSE OF LMDER1 IS TO MINIMIZE THE SUM OF THE SQUARES OF
C     M NONLINEAR FUNCTIONS IN N VARIABLES BY A MODIFICATION OF THE
C     LEVENBERG-MARQUARDT ALGORITHM. THIS IS DONE BY USING THE MORE
C     GENERAL LEAST-SQUARES SOLVER LMDER. THE USER MUST PROVIDE A
C     SUBROUTINE WHICH CALCULATES THE FUNCTIONS AND THE JACOBIAN.
C
C INPUT:
C       FCN IS THE NAME OF THE USER-SUPPLIED SUBROUTINE WHICH
C         CALCULATES THE FUNCTIONS AND THE JACOBIAN. FCN MUST
C         BE DECLARED IN AN EXTERNAL STATEMENT IN THE USER
C         CALLING PROGRAM, AND SHOULD BE WRITTEN AS FOLLOWS.
C
C         SUBROUTINE FCN(M,N,X,FVEC,FJAC,LDFJAC,IFLAG)
C         INTEGER M,N,LDFJAC,IFLAG
C         DOUBLE PRECISION X(N),FVEC(M),FJAC(LDFJAC,N)
C         ----------
C         IF IFLAG = 1 CALCULATE THE FUNCTIONS AT X AND
```

```
C          RETURN THIS VECTOR IN FVEC. DO NOT ALTER FJAC.
C          IF IFLAG = 2 CALCULATE THE JACOBIAN AT X AND
C          RETURN THIS MATRIX IN FJAC. DO NOT ALTER FVEC.
C          ----------
C          RETURN
C          END
C
C          THE VALUE OF IFLAG SHOULD NOT BE CHANGED BY FCN UNLESS
C          THE USER WANTS TO TERMINATE EXECUTION OF LMDER1.
C          IN THIS CASE SET IFLAG TO A NEGATIVE INTEGER.
C
C        M IS A POSITIVE INTEGER INPUT VARIABLE SET TO THE NUMBER
C          OF FUNCTIONS.
C
C        N IS A POSITIVE INTEGER INPUT VARIABLE SET TO THE NUMBER
C          OF VARIABLES. N MUST NOT EXCEED M.
C
C        LDFJAC IS A POSITIVE INTEGER INPUT VARIABLE NOT LESS THAN M
C
C          WHICH SPECIFIES THE LEADING DIMENSION OF THE ARRAY FJAC.
C
CC        TOL IS A NONNEGATIVE INPUT VARIABLE. TERMINATION OCCURS
CC          WHEN THE ALGORITHM ESTIMATES EITHER THAT THE RELATIVE
CC          ERROR IN THE SUM OF SQUARES IS AT MOST TOL OR THAT
CC          THE RELATIVE ERROR BETWEEN X AND THE SOLUTION IS AT
CC          MOST TOL.
C
C        WA IS A DOUBLE PREC. WORK ARRAY OF LENGTH LWA.
C
C        LWA IS A POSITIVE INTEGER INPUT VARIABLE NOT LESS THAN 5*N+M.
C
C I/O:
C        X IS AN ARRAY OF LENGTH N. ON INPUT X MUST CONTAIN
C          AN INITIAL ESTIMATE OF THE SOLUTION VECTOR. ON OUTPUT X
C          CONTAINS THE FINAL ESTIMATE OF THE SOLUTION VECTOR.
C
C OUTPUT:
C        FVEC IS AN OUTPUT ARRAY OF LENGTH M WHICH CONTAINS
C          THE FUNCTIONS EVALUATED AT THE OUTPUT X.
C
C        FJAC IS AN OUTPUT M BY N ARRAY. THE UPPER N BY N SUBMATRIX
C          OF FJAC CONTAINS AN UPPER TRIANGULAR MATRIX R WITH
C          DIAGONAL ELEMENTS OF NONINCREASING MAGNITUDE SUCH THAT
C
C                 T     T           T
C               P *(JAC *JAC)*P = R *R,
C
C          WHERE P IS A PERMUTATION MATRIX AND JAC IS THE FINAL
C          CALCULATED JACOBIAN. COLUMN J OF P IS COLUMN IPVT(J)
C          (SEE BELOW) OF THE IDENTITY MATRIX. THE LOWER TRAPEZOIDAL
C          PART OF FJAC CONTAINS INFORMATION GENERATED DURING
C          THE COMPUTATION OF R.
C
C        INFO IS AN INTEGER OUTPUT VARIABLE. IF THE USER HAS
C          TERMINATED EXECUTION, INFO IS SET TO THE (NEGATIVE)
C          VALUE OF IFLAG. SEE DESCRIPTION OF FCN. OTHERWISE,
C          INFO IS SET AS FOLLOWS.
C
C          INFO = 0  IMPROPER INPUT PARAMETERS.
C
C          INFO = 1  ALGORITHM ESTIMATES THAT THE RELATIVE ERROR
C                    IN THE SUM OF SQUARES IS AT MOST TOL.
C
C          INFO = 2  ALGORITHM ESTIMATES THAT THE RELATIVE ERROR
C                    BETWEEN X AND THE SOLUTION IS AT MOST TOL.
C
C          INFO = 3  CONDITIONS FOR INFO = 1 AND INFO = 2 BOTH HOLD.
C
C          INFO = 4  FVEC IS ORTHOGONAL TO THE COLUMNS OF THE
C                    JACOBIAN TO MACHINE PRECISION.
C
C          INFO = 5  NUMBER OF CALLS TO FCN WITH IFLAG = 1 HAS
C                    REACHED 100*(N+1).
C
C          INFO = 6  TOL IS TOO SMALL. NO FURTHER REDUCTION IN
C                    THE SUM OF SQUARES IS POSSIBLE.
C
C        IPVT IS AN INTEGER OUTPUT ARRAY OF LENGTH N. IPVT
C          DEFINES A PERMUTATION MATRIX P SUCH THAT JAC*P = Q*R,
C          WHERE JAC IS THE FINAL CALCULATED JACOBIAN, Q IS
C          ORTHOGONAL (NOT STORED), AND R IS UPPER TRIANGULAR
C          WITH DIAGONAL ELEMENTS OF NONINCREASING MAGNITUDE.
C          COLUMN J OF P IS COLUMN IPVT(J) OF THE IDENTITY MATRIX.
C
C CALLING PROGRAM: GENERAL USE
C
C ROUTINES:
C----------------------------------------------------------------------
C     NAME     SOURCE   PURPOSE
C----------------------------------------------------------------------
C     FCN      USER     SEE ABOVE
C     LMDER    MINPACK  DOES THE CALCULATION. FOLLOWS LATER IN THIS FILE
C----------------------------------------------------------------------
C
C AUTHOR: ARGONNE NATIONAL LABORATORY. MINPACK PROJECT. MARCH 1980.
C     BURTON S. GARBOW, KENNETH E. HILLSTROM, JORGE J. MORE
C
C DATE:   05-06-96
```

```
C
C VERSION 1.1                              DATE: 05-06-96
C MODIFIED: WILLIAM OSBORN
C           - FIRST COPIED FOR ADAS USE. REMOVED TOL FROM PARAMETERS
C             AND HARDWIRED AS DSQRT(DPMPAR(1))
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------

      integer m,n,ldfjac,info,lwa
      integer ipvt(n)
      double precision x(n),fvec(m),fjac(ldfjac,n),wa(lwa)
      external fcn
c     **********
c
c     subroutine lmder1
c
c     **********
      integer maxfev,mode,nfev,njev,nprint
      double precision factor,ftol,gtol,xtol,zero,tol, dpmpar
      data factor,zero /1.0d2,0.0d0/
C-----------------------------------------------------------------------
```

XXDSN2

```
      SUBROUTINE XXDSN2 ( DSNIN   ,
     &                    DSNOUT  , LEXIST
     &                  )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXDSN2 *********************
C
C  PURPOSE: TO CHECK IF A FULLY QUALIFIED MVS SEQUENTIAL OR PARTITIONED
C           DATA SET EXISTS AND RETURNS THE DATA SET NAME IN A FORM FOR
C           DYNAMIC ALLOCATION.
C
C                   ***** 'DSNIN' MUST BE OF VALID FORM *****
C
C           NOTE: INPUT DSN MUST START WITH THE USERID AND MAY OR MAY
C                 NOT BE ENCLOSED IN QUOTES.  IT CAN ALREADY BE  IN A
C                 FORM SUITABLE FOR DYNAMIC  ALLOCATION IN WHICH CASE
C                 THE OUTPUT DSN AND INPUT DSN  ARE  IDENTICAL   (I.E.
C                 NEITHER HAVE QUOTES AND BOTH START WITH A '/'.)
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (C*(*)) DSNIN   = DATASET NAME
C                            (CAN BE ENCLOSED WITHIN APOSTROPHES OR
C                             START WITH A '/')
C
C  OUTPUT: (C*(*)) DSNOUT  = VERSION OF  'DSNIN'  ALETRED SO THAT IT
C                            CAN BE USED TO DYNAMICALLY OPEN THE FILE.
C                            (NOTE: APOSTROPHES REMOVED, / ADDED AT
C                             START IF REQUIRED.)
C  OUTPUT: (L*4)   LEXIST  = .TRUE.  => FILE 'DSNIN' ALREADY EXISTS
C                          = .FALSE. => FILE 'DSNIN' DOES NOT EXISTS
C
C          (C*1)   QUOTE   = PARAMETER: APOSTROPHE (').
C          (C*1)   BSLASH  = PARAMETER: BACK SLASH (/).
C
C          (I*4)   ILEND   = LENGTH IN BYTES OF 'DSNIN'
C          (I*4)   I1      = LOCATION OF FIRST APOSTROPHE IN 'DSNIN'
C          (I*4)   I2      = LOCATION OF SECOND APOSTROPHE IN 'DSNIN'
C          (I*4)   IDSLEN  = IDENTIFIES THE NUMBER OF CHARACTERS PRESENT
C                            IN THE INPUT DATA SET NAME MINUS ONE.
C                            (I.E. NUMBER   OF   CHARACTERS  BETWEEN
C                                  APOSTROPHES IF PRESENT.)
C
C          (L*4)   LQUOTE  = .TRUE.  => APOSTROPHES PRESENT
C                            .FALSE. => NO APOSTROPHES PRESENT
C
C NOTE:
C
C          IDSLEN >=0 => NO APOSTROPHES IN 'DSNIN' OR TWO SEPERATED
C                        BY AT LEAST ONE CHARACTER.
C          IDSLEN =-1 => TWO ADJACENT APOSTROPHES
C          IDSLEN =-2 => ONLY ONE APOSTROPHE FOUND
C
C          IF:
C
C          IDSLEN =-1 ERROR: PROGRAM STOPS
C          IDSLEN =-2 ERROR: PROGRAM STOPS.
C
C
C ROUTINES:   NONE
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    12/12/90  (IN ADAS90 VERSION 1 THIS ROUTINE IS CALLED XXDSN1)
C
```

```
C-----------------------------------------------------------------------
      CHARACTER  QUOTE*1     , BSLASH*1
C-----------------------------------------------------------------------
      PARAMETER( QUOTE=''''  , BSLASH='/' )
C-----------------------------------------------------------------------
      INTEGER    IDSLEN      , I1         , I2        , ILEND
C-----------------------------------------------------------------------
      CHARACTER  DSNIN*(*)   , DSNOUT*(*)
C-----------------------------------------------------------------------
      LOGICAL    LEXIST      , LQUOTE
C-----------------------------------------------------------------------
```

XXDTES

```
      SUBROUTINE XXDTES( CSTR19 , LEISS  , LSTAN  , NVLCE )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXDTES *********************
C
C  PURPOSE: DETECTS IF THE CONFIGURATION STRING FROM A SPECIFIC ION
C           LEVEL LIST LINE IS OF EISSNER OR STANDARD FORM
C
C  CALLING PROGRAMS: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (C*19)  CSTR19  = CONFIGURATION CHARACTER STRING
C  OUTPUT: (L*4)   LEISS   = .TRUE.  => EISSNER FORM
C                            .FALSE. => NOT EISSNER FORM
C  OUTPUT: (L*4)   LSTAN   = .TRUE.  => STANDARD FORM
C                            .FALSE. => NOT STANDARD FORM
C  OUTPUT: (I*4)   NVLCE   = OUTER ELECTRON N-SHELL IF RECOGNISABLE
C
C          (I*4)   I       = GENERAL USE
C          (I*4)   IABT    = RETURN CODE (SEE SPECIFIC FUNCTION)
C                            0 => OK
C                            1 => FAULT DETECTED
C          (I*4)   ICFSEL  = 1 => STANDARD FORM OUT, STANDARD FORM IN
C                            2 => EISSNER  FORM OUT, STANDARD FORM IN
C                            3 => STANDARD FORM OUT, EISSNER  FORM IN
C                            4 => EISSNER  FORM OUT, EISSNER  FORM IN
C          (I*4)   ISHEL   = SHELL COUNTER
C          (I*4)   IP      = PARITY OF CONFIGURATION
C          (I*4)   MAXN    = N_SHELL SUM FOR CONFIGURATION
C          (I*4)   NSHEL   = NUMBER OF SHELLS IDENTIFIED fFROM STRING
C          (I*4)   NDWORD  = MAXIMUM NUMBER OF WORDS IN STRING
C          (I*4)   NFIRST  = FIRST WORD TO BE EXTRACTED FROM STRING
C          (I*4)   NWORDS  = NUMBER OF WORDS IN STRING
C          (I*4)   NELA()  = NUMBER OF ELECTRONS IN EACH SHELL
C          (I*4)   IFIRST()= POSITION OF FIRST CHAR. OF WORD IN STRING
C          (I*4)   ILAST() = POSITION OF LAST CHAR. OF WORD IN STRING
C
C          (C*1)   CDELIM  = SEPARATORS FOR WORDS IN STRING
C          (C*19)  CSTRGO  = GENERAL USE STRING
C          (C*19)  STRG    = STANDARD FORM CONFIGURATION STRING
C          (C*19)  STRGE   = EISSNER FORM CONFIGURATION STRING
C          (C*1)   CHEISA()= EISSNER CHARACTER FOR ORBITALS
C          (C*2)   CHSTDA()= STANDARD ORBITAL SPEC. FOR EACH SHELL
C          (C*2)   CNELA() = CHARS. FOR NO. OF EQUIV. ELEC. IN SHELL
C                            (EISSNER FORM CASE)
C          (C*1)   CHQA()  = INDEX TO HEXADECIMAL CONVERSIONS
C          (C*1)   CHRA()  = CHAR. FOR NO. OF. EQUIV. ELEC. IN SHELL
C                            (STANDARD FORM CASE)
C
C ROUTINES:
C          ROUTINE    SOURCE   BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4FCTN     ADAS     CONVERTS CHARACTER STRING TO INTEGER
C          I4NGRP     ADAS     RETURNS N QUANTUM NUMBER IN THE
C                              EISSNER SINGLE HEXADECIMAL CHARACTER FORM
C          I4PGRP     ADAS     RETURNS PARITY OF ORBITAL GIVEN THE
C                              EISSNER SINGLE HEXADECIMAL CHARACTER FORM
C          I4SCHR     ADAS     RETURNS NUMERICAL VALUE FOR NUMBER OF
C                              EQUIVALENT ELECTRONS GIVEN AS HEX> CHAR.
C          CSTGRP     ADAS     RETURNS TERM OF ORBITAL GIVEN IN THE
C                              EISSNER SINGLE HEXADECIMAL CHARACTER FORM
C          CEIGRP     ADAS     RETURNS EISSNER CODE FOR ORBITAL
C          XXWORD     ADAS     FINDS NUMBER OF WORDS IN A STRING
C
C
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE:    01/11/95
C
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 19-1-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER SCCS CONTROL
C
```

```
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER   NDWORD
C-----------------------------------------------------------------------
      PARAMETER ( NDWORD = 6 )
C-----------------------------------------------------------------------
      INTEGER   NVLCE     , I         , NFIRST  , IWORDS   , NWORDS
      INTEGER   I4UNIT    , I4NGRP    , I4PGRP  , I4SCHR   , I4FCTN
      INTEGER   ISHEL     , NSHEL     , IP      , MAXN     , ICFSEL
      INTEGER   IABT
C-----------------------------------------------------------------------
      CHARACTER CSTRGO*19 , STRG*19   , STRGE*19
      CHARACTER CSTR19*19 , CDELIM*1
      CHARACTER CSTGRP*2  , CEIGRP*1
C-----------------------------------------------------------------------
      LOGICAL*4 LEISS     , LSTAN
C-----------------------------------------------------------------------
      INTEGER   NELA(6)   , IFIRST(NDWORD)      , ILAST(NDWORD)
C-----------------------------------------------------------------------
      CHARACTER CHEISA(6)*1 , CHSTDA(4)*2  , CNELA(6)*2
      CHARACTER CHQA(15)*1  , CHRA(4)*1
C-----------------------------------------------------------------------
```

XXECON

```
      SUBROUTINE XXECON( INTYP, OUTTYP, IEVAL, EIN, EOUT )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXECON *********************
C
C  PURPOSE: TO CONVERT AN ARRAY OF VELOCITIES/ENERGIES INTO A  SPECIFIED
C           FORM.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT :    (I*4)  INTYP   = 1 => 'EIN(array)' UNITS: AT. UNITS (VEL.)
C                              = 2 => 'EIN(array)' UNITS: CM/SEC    (VEL.)
C                              = 3 => 'EIN(array)' UNITS: EV/AMU  (ENERGY)
C  INPUT :    (I*4)  OUTTYP  = 1 => 'EOUT(array)' UNITS: AT.UNITS (VEL.)
C                              = 2 => 'EOUT(array)' UNITS: CM/SEC   (VEL.)
C                              = 3 => 'EOUT(array)' UNITS: EV/AMU (ENERGY)
C  INPUT :    (I*4)  IEVAL   = NO. OF VELOCITIES/ENERGIES IN EIN(array)
C  INPUT :    (R*8)  EIN()   = INPUT  VELOCITIES/ENERGIES (STATED UNITS)
C  OUTPUT:    (R*8)  EOUT()  = OUTPUT VELOCITIES/ENERGIES (STATED UNITS)
C
C             (R*8)  AMU2KG  = PARAMETER: AMU TO KG CONVERSION FACTOR
C             (R*8)  EV2J    = PARAMETER: EV TO JOULES CONVERSION FACTOR
C             (R*8)  M2CM    = PARAMETER: METRES TO CM CONVERSION FACTOR
C             (R*8)  VELE0H  = PARAMETER: ORBITAL VELOCITY  (CM/SEC)  OF
C                              AN ELECTRON IN THE SMALLEST  ORBIT  OF  A
C                              HYDROGEN ATOM (BOHR) = 2.1877D+8 CM/SEC
C             (R*8)  AT2VEL  = AT.UNITS (VEL) TO CM/SEC (VEL) CONVERSION
C             (R*8)  VEL2AT  = CM/SEC (VEL) TO AT.UNITS (VEL) CONVERSION
C             (R*8)  VEL2EN  = CM/SEC (VEL) TO EV/AMU (ENGY.) CONVERSION
C             (R*8)  EN2VEL  = EV/AMU (ENGY.) TO CM/SEC (VEL) CONVERSION
C             (R*8)  AT2EN   = AT.UNITS (VEL) TO EV/AMU (ENG) CONVERSION
C             (R*8)  EN2AT   = EV/AMU (ENG) TO AT.UNITS (VEL) CONVERSION
C
C             (I*4)  I       = GENERAL USE
C
C             (R*8)  ECONV() = ENERGY/VELOCITY CONVERSION PARAMETERS
C
C ROUTINES:  NONE
C
C NOTE:
C            ENERGY/VELOCITY CONVERSION PARAMETERS:
C
C            INTYP = 1 ; ECONV(1) =>  VELOCITY: AT.UNITS -> OUTPUT FORM
C            INTYP = 2 ; ECONV(2) =>  VELOCITY: CM/SEC   -> OUTPUT FORM
C            INTYP = 3 ; ECONV(3) =>  ENERGY  : EV/AMU   -> OUTPUT FORM
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    05/02/91
C
C VERSION: 1.2                                      DATE: 01-05-96
C MODIFIED: WILLIAM OSBORN
C            - CORRECTED AT2EN. IT WAS AT2VEL*VEL2EN.
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8     AMU2KG                 , EV2J                  ,
     &           M2CM                   , VELE0H
      REAL*8     AT2VEL                 , VEL2AT
      REAL*8     VEL2EN                 , EN2VEL
      REAL*8     AT2EN                  , EN2AT
C-----------------------------------------------------------------------
      PARAMETER( AMU2KG = 1.66053D-27    , EV2J   = 1.602192D-19  ,
     &           M2CM   = 1.0D+02        , VELE0H = 2.1877D+08    )
      PARAMETER( AT2VEL = VELE0H         , VEL2AT = 1.0/VELE0H    )
```

```
      PARAMETER( VEL2EN = AMU2KG/( 2.0*EV2J*M2CM*M2CM )
     &           EN2VEL = ( 2.0*EV2J*M2CM*M2CM )/AMU2KG          )
      PARAMETER( AT2EN  = AT2VEL*AT2VEL*VEL2EN                   ,
     &           EN2AT  = EN2VEL*VEL2AT*VEL2AT                   )
C-----------------------------------------------------------------------
      INTEGER    INTYP                , OUTTYP                  ,
     &           IEVAL
      INTEGER    I
C-----------------------------------------------------------------------
      REAL*8     EIN(IEVAL)           , EOUT(IEVAL)
      REAL*8     ECONV(3)
C-----------------------------------------------------------------------
```

XXEIGN

```
      SUBROUTINE XXEIGN(A,IA,N,RR,RI,VR,VI,IV1,FV1,IERR)
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C ROUTINE: XXEIGN
C
C PURPOSE: FINDS THE EIGENVALUES AND EIGENVECTORS OF A GENERAL REAL
C          MATRIX. REPLACES NAG ROUTINE F02AGF ALTHOUGH THERE ARE SEVERAL
C          DIFFERENCES - YOU SHOULD COMPARE THE DOCUMENTATION.
C          THIS IS A FRONT-END TO THE NETLIB LIBRARY PROGRAM RG.F WHICH
C          FOLLOWS TOGETHER WITH ALL ITS DEPENDENCIES.
C*** N.B. THE EIGENVECTORS ARE NOT NORMALIZED TO UNIT LENGTH ***
C
C CALLING PROGRAM: GENERAL USE
C
C INPUT:
C
C     (R*8)   A(,)    CONTAINS THE REAL GENERAL MATRIX.
C
C     (I*4)   IA      THE ROW DIMENSION OF THE TWO-DIMENSIONAL
C                     ARRAY PARAMETERS AS DECLARED IN THE CALLING
C                     PROGRAM DIMENSION STATEMENT.
C
C     (I*4)   N       THE ORDER OF THE MATRIX  A.
C     (I*4)   IV1()   WORK ARRAY, DIMENSION = N.
C     (R*4)   FV1()   WORK ARRAY, DIMENSION = N.
C
C OUTPUT:
C
C     (R*8)   RR()    REAL PART OF THE EIGENVALUES
C     (R*8)   RI()    IMAGINARY PART OF EIGENVALUES
C                     COMPLEX CONJUGATE PAIRS OF EIGENVALUES APPEAR
C                     CONSECUTIVELY WITH THE EIGENVALUE HAVING THE
C                     POSITIVE IMAGINARY PART FIRST.
C
C     (R*8)   VR(,)   REAL PARTS OF THE EIGENVECTORS. FIRST DIMENSION
C                     = IA.
C                     THE REAL PART OF THE EIGENVECTOR CORRESPONDING TO
C                     THE I-TH EIGENVALUE IS IN VR(J,I), J=1...N
C     (R*8)   VI(,)   IMAGINARY PARTS OF THE EIGENVECTORS. FIRST
C                     DIMENSION = IA
C                     THE IMAGINARY PART OF THE EIGENVECTOR
C                     CORRESPONDING TO THE I-TH EIGENVALUE IS IN
C                     VR(J,I), J=1...N
C     (I*4)   IERR    AN ERROR CODE SET TO ZERO IF NO ERROR. SEE HRQ AND
C                     HRQ2 FOR A DESCRIPTION OF THIS VARIABLE
C
C ROUTINES:
C-----------------------------------------------------------------------
C     NAME     SOURCE   PURPOSE
C-----------------------------------------------------------------------
C     RG       NETLIB   CALCULATES THE EIGENVALUES AND EIGENVECTORS
C     DPMPAR   ADAS/NETLIB   RETURNS MACHINE DEPENDANT PARAMETERS
C-----------------------------------------------------------------------
C
C AUTHOR: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC.)
C
C DATE:   10/06/96
C
C VERSION 1.1                           DATE: 10/06/96
C MODIFIED: WILLIAM OSBORN
C           - FIRST VERSION
C VERSION: 1.2                                DATE:27/06/96
C MODIFIED: WILLIAM OSBORN
C            - REMOVED UNUSED VARIABLES
C
C-----------------------------------------------------------------------
C
      INTEGER IA,N,IERR
      REAL*8 A(IA,N), RR(N),RI(N), VR(IA,N), VI(IA,N), FV1(N)
      INTEGER IV1(N)
      REAL*8 ZEROP, DPMPAR
      INTEGER MATZ,I,J
C-----------------------------------------------------------------------
```

XXEIZ0

```
      SUBROUTINE XXEIZ0 ( ESYM , IZ0 )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C ******************* FORTRAN77 SUBROUTINE: XXEIZ0 *********************
C
C PURPOSE: TO RETURN THE NUCLEAR CHARGE IZ0 FOR THE ELEMENT SYMBOL ESYM
C
C CALLING PROGRAM: GENERAL USE
C
C SUBROUTINE:
C
C  INPUT : (C*2)  ESYM    = SYMBOL OF ELEMENT WITH NUCLEAR CHARGE 'IZ0'
C  OUTPUT: (I*4)  IZ0     = ELEMENT NUCLEAR CHARGE
C
C          (I*4)  NSYM    = PARAMETER = NUMBER OF SYMBOLS LISTED
C
C          (I*4)  I       = GENERAL ARRAY USE
C
C          (C*2)  SYMBOL()= SYMBOLS OF FIRST 'NSYM' ELEMENTS.
C                           ARRAY DIMENSION => NUCLEAR CHARGE
C
C NOTES:    IF SYMBOL IS NOT RECOGNISED, I.E.NOT IN Z0 RANGE 1 & 'NSYM',
C           THEN THE INTEGER 'IZ0' IS RETURNED AS ZERO.
C
C ROUTINES: NONE
C
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     21/08/90
C
C DATE:     15/01/91 - PE BRIDEN - ADAS91 - ARGUMENTS REVERSED
C
C MODIFIED:
C
C VERSION:  1.1  TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)  14/09/95
C            -   FIRST RELEASE (UNIX PORT)
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER    NSYM
C-----------------------------------------------------------------------
      PARAMETER( NSYM = 50 )
C-----------------------------------------------------------------------
      INTEGER    IZ0       , I
C-----------------------------------------------------------------------
      CHARACTER  ESYM*2    , SYMBOL(NSYM)*2
C-----------------------------------------------------------------------
```

XXELEM

```
      SUBROUTINE XXELEM ( IZ0 , ENAME )
IMPLICIT NONE
C-----------------------------------------------------------------------
C
C ******************* FORTRAN77 SUBROUTINE: XXELEM *********************
C
C PURPOSE: TO RETURN THE NAME OF THE ELEMENT WITH NUCLEAR CHARGE IZ0
C
C CALLING PROGRAM: GENERAL USE
C
C SUBROUTINE:
C
C  INPUT : (I*4)  IZ0     = ELEMENT NUCLEAR CHARGE
C  OUTPUT: (C*12) ENAME   = NAME OF ELEMENT WITH NUCLEAR CHARGE 'IZ0'
C
C          (C*12) NAMES() = NAMES OF FIRST 50 ELEMENTS.
C                           ARRAY DIMENSION => NUCLEAR CHARGE
C
C NOTES:    IF NUCLEAR CHARGE IS OUT OF RANGE, I.E.NOT BETWEEN 1 & 50,
C           THEN THE CHARACTER STRING 'ENAME' IS RETURNED BLANK.
C
C ROUTINES: NONE
C
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     05/07/90
C
C UPDATE:   15/01/91 - PE BRIDEN - ADAS91 - ARGUMENTS REVERSED
C
C UPDATE:   25/01/91 - PE BRIDEN - ADAS91 - 'ENAME' & 'NAMES()' DECLARED
C                                           AS C*12 INSTEAD OF C*15.
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 21-03-96
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - PUT UNDER S.C.C.S. CONTROL
C
```

```
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER      IZ0
C-----------------------------------------------------------------------
      CHARACTER*12 ENAME , NAMES(50)
C-----------------------------------------------------------------------
```

XXERYD

```
      SUBROUTINE XXERYD( BWNO , IL , WA  ,
     &                        ER , XIA
     &                 )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXERYD *********************
C
C  PURPOSE: TO CALCULATE THE ENERGY LEVELS IN RYDBERGS ( FROM WAVE NUM-
C           BERS) RELATIVE TO LEVEL 1, AND THE ENERGIES (ALSO IN RYD.)
C           RELATIVE TO THE IONISATION POTENTIAL.
C
C  CALLING PROGRAM:  GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT :    (I*4)  BWNO    = IONISATION POTENTIAL (CM-1)
C  INPUT :    (I*4)  IL      = NUMBER OF ENERGY LEVELS
C  INPUT :    (R*8)  WA()    = ENERGY RELATIVE TO LEVEL 1 (CM-1)
C
C  OUTPUT:    (R*8)  ER()    = ENERGY RELATIVE TO LEVEL 1 (RYDBERGS)
C  OUTPUT:    (R*8)  XIA()   = ENERGY RELATIVE TO ION. POT. (RYDBERGS)
C
C             (R*8)  WN2RYD  = WAVE NUMBER (CM-1) TO RYDBERG CONVERSION
C                              (PARAMETER)
C
C             (I*4)  I       = GENERAL USE
C
C ROUTINES:  NONE
C
C NOTES:     ION. POT. = IONISATION POTENTIAL
C
C AUTHOR:    PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C            K1/0/81
C            JET EXT. 4569
C
C DATE:      05/07/90
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER IL     , I
C-----------------------------------------------------------------------
      REAL*8  WN2RYD
      REAL*8  BWNO
      REAL*8  WA(IL) , ER(IL) , XIA(IL)
C-----------------------------------------------------------------------
```

XXFCHR

```
      SUBROUTINE XXFCHR( CSTRNG , SSTRNG, IFIRST , ILAST )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXFCHR *********************
C
C  PURPOSE: TO IDENTIFY THE FIRST AND LAST OCCURRENCE OF SSTRNG IN
C           CSTRNG, THE VALUES OF WHICH ARE IFIRST , ILAST.
C           - IF    NO    OCCURRENCE OF SSTRNG THEN IFIRST=ILAST=0
C           - IF ONLY ONE OCCURRENCE OF SSTRNG THEN IFIRST=ILAST>0
C           - IF OVER ONE OCCURRENCE OF SSTRNG THEN IFIRST>ILAST>0
C
C  NOTE   : ANY TRAILING BLANKS IN THE SEARCH STRING (SSTRNG) ARE
C           IGNORED.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (C*(*)) CSTRNG   = INPUT STRING FOR INTERROGATION
C  INPUT : (C*(*)) SSTRNG   = INPUT SEARCH STRING
C
C  OUTPUT: (I*4)   IFIRST   = BYTE POSITION OF FIRST OCCURRENCE OF SSTRNG
C                             IN CSTRNG.
C  OUTPUT: (I*4)   ILAST    = BYTE POSITION OF LAST  OCCURRENCE OF SSTRNG
C                             IN CSTRNG.
C
C          (I*4)   ILENC    = LENGTH OF 'CSTRNG' STRING IN BYTES
C          (I*4)   ILENS    = POSITION OF LAST  NON-BLANK BYTE IN SSTRNG
C          (I*4)   I        = GENERAL USE - INCLUDING DUMP FOR UNWANTED
C                             OUTPUT FROM XXSLEN SUBROUTINE.
C
C ROUTINES:
```

```
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSLEN     ADAS      FIND FIRST/LAST NONBLANK BYTES IN STRING
C
C NOTE  :  ANY TRAILING BLANKS IN THE SEARCH STRING (SSTRNG) ARE
C          IGNORED.
C
C
C AUTHOR:  LALIT JALOTA (TESSELLA SUPPORT SERVICES PLC)
C
C DATE  :  27/10/94
C
C UPDATE:  06/03/95 - REVISED BY PAUL BRIDEN (TESSELLA SUPPORT SERVICES)
C                     1) ANALYSE ALL OF CSTRNG (NOT JUST NON-BLANK
C                        PART).
C                     2) ONLY IGNORE TRAILING BLANKS FOR SSTRNG
C                        (KEEP LEADING BLANKS).
C                     3) MODIFY DO LOOP INDEX RANGE TO ENSURE THAT
C                        YOU DO NOT GO BEYOND THE END OF CSTRNG.
C                     4) VERIFY LENGTH OF SSTRNG IS NON-ZERO.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER     IFIRST     , ILAST
      INTEGER     ILENC      , ILENS      , I
C-----------------------------------------------------------------------
      CHARACTER   CSTRNG*(*) , SSTRNG*(*)
C-----------------------------------------------------------------------
```

XXFLNM

```
      SUBROUTINE XXFLNM( DSNIN  , DSNFUL , LEXIST )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXFLNM *********************
C
C  PURPOSE: TO PREPARE A UNIX DATASET NAME FROM A STRING WHICH MAY
C           INCLUDE AN ADAS ENVIRONMENT LEADER AND COMMENTS.
C           THE ADAS ENVIRONMENT VARIABLE MUST BE FIRST AND IN DOUBLE
C           QUOTES.  THE COMMENTS MUST EITHER FOLLOW OR PRECEDE A COLON.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (C80)  DSNIN    = INPUT STRING FOR INTERROGATION
C
C  OUTPUT: (C80)  DSNFUL   = THE FULL EXPANDED FILE NAME WITHOUT
C                            EXTRANEOUS MATERIAL
C  OUTPUT: (L*4)  LEXIST   = .TRUE. => NAME FORMED AND FILE EXISTS
C                            .FALSE.=> FAILED TO FORM NAME OR FIND FILE
C
C          (C*80) DSN1     = WORK STRING
C          (C*80) DSNTEMP  = WORK STRING
C          (C*80) BLANK    = BLANK STRING
C          (I*4)   LEN1     = STRING INDEX
C          (I*4)   LEN2     = STRING INDEX
C          (I*4)   LEN3     = STRING INDEX
C          (I*4)   LEN4     = STRING INDEX
C
C ROUTINES:
C          ROUTINE    SOURCE   BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXSLEN     ADAS     FIND BEGINNING AND END OF A STRING
C          I4UNIT     ADAS     FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          GETENV     UNIX     FETCH AN ENVIRONMENT VARIABLE
C
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE  :  21/05/96
C
C UPDATE:
C
C VERSION: 1.1                          DATE: 29-05-96
C MODIFIED: WILLIAM OSBORN
C               - ADDED CODE FOR INITIAL COMMENTS AND PUT INTO S.C.C.S.
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C

      INTEGER I4UNIT
      INTEGER LEN1, LEN2, LEN3, LEN4, P1, P2
      CHARACTER DSNIN*80, DSNFUL*80, DSN1*80, BLANK*80, DSNTEMP*80
      LOGICAL LEXIST
C-----------------------------------------------------------------------
```

XXFLSH

```
      SUBROUTINE XXFLSH(PIPEOU)
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXFLSH *********************
C
C  PURPOSE: ROUTINE FOR SETTING CALL TO "FLUSH" COMMAND DEPENDING ON
C           OPERATING SYSTEM SPCIFICS.
C
C  CALLING PROGRAM: ANY ADAS PROGRAM WHICH COMMUNICATES VIA A PIPE
C                   PARTICULARLY TO IDL INTERFACE ROUTINES.
C
C
C  SUBROUTINE:
C
C          (I*4)  PIPEOU  = UNIT NUMBER OF PIPE USED
C
C NOTE:
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C AUTHOR:  L. JALOTA (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:     6/03/95
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER    PIPEOU
C-----------------------------------------------------------------------
```

XXGAMA

```
      SUBROUTINE XXGAMA( MXINDX , JGAM, GAM )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXGAMA *********************
C
C  PURPOSE: SET UP GAMMA FUNCTION TABLES FOR INTEGER ARGUMENTS
C
C           TABLES 'JGAM' AND 'GAM' CAN BE REFERENCED WITH FUNCTIONS
C           'I4JGAM' AND 'R8GAM', RESPECTIVELY.
C
C  CALLING PROGRAM: GENERAL USE.
C
C  INPUT : (I*4)  MXINDX  =
C
C  OUTPUT: (I*4)  JGAM()  =
C                           DIMENSION: REFERENCED BY I.
C  OUTPUT: (I*4)  GAM()   =
C                           DIMENSION: REFERENCED BY I.
C
C  PARAM : (R*8)  P1      = 64.0
C
C          (I*4)  I       = LOOP INDEX.
C          (I*4)  I1      = I-1
C          (I*4)  J1      = USED IN COMPUTATION ( = JGAM(N) ).
C
C          (R*8)  X1      = USED IN COMPUTATION ( = GAM(N) ).
C
C ROUTINES: NONE
C
C AUTHOR:   JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 5183
C
C DATE:     30/09/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8     P1
      PARAMETER( P1 = 64.00D+00 )
C-----------------------------------------------------------------------
      INTEGER    MXINDX
      INTEGER    I       , I1      , J1
C-----------------------------------------------------------------------
      REAL*8     X1
C-----------------------------------------------------------------------
      INTEGER    JGAM(MXINDX)
C-----------------------------------------------------------------------
      REAL*8     GAM(MXINDX)
C-----------------------------------------------------------------------
```

XXGTSL

```
      SUBROUTINE XXGTSL( N  , D  , DU , DL , B  , IFAIL )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXGTSL *********************
C
C  PURPOSE: GIVEN A GENERAL TRIDIAGONAL MATRIX AND A RIGHT HAND SIDE
C           WILL FIND THE SOLUTION OF THE ASSOCIATED SYSTEM OF LINEAR
C           EQUATIONS.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C  INPUT :    (I*4)  N       = ORDER OF TRIDIAGONAL MATRIX
C
C  INPUT :    (R*8)  DL()    = SUBDIAGONAL OF THE MATRIX.  DL(2) THROUGH
C                              DL(N) SHOULD CONTAIN THE SUBDIAGONAL. ON
C                              OUTPUT DL IS DESTROYED
C  INPUT :    (R*8)  D()     = DIAGONAL OF THE MATRIX.  ON OUTPUT D
C                              IS DESTROYED
C  INPUT :    (R*8)  DU()    = SUPERDIAGONAL OF THE MATRIX. DU(2) THROUGH
C                              DU(N) SHOULD CONTAIN THE SUPERDIAGONAL.
C                              OUTPUT DU IS DESTROYED
C  INPUT :    (R*8)  B()     = RIGHT HAND SIDE VECTOR
C
C  OUTPUT:    (R*8)  B()     = SOLUTION VECTOR
C
C  OUTPUT:    (I*4)  IFAIL   = 0 - NORMAL VALUE
C                              = K - IF THE KTH PIVOT ELEMENT BECOMES
C                                    BECOMES EXACTLY ZERO.  THE ROUTINE
C                                    RETURNS WHEN THIS IS THE CASE.
C
C             (I*4)  K       = GENERAL INTEGER
C             (I*4)  KB      = GENERAL INTEGER
C             (I*4)  KP1     = K+1
C             (I*4)  NM1     = N-1
C             (I*4)  NM2     = N-2
C             (R*8)  T       = GENERAL REAL
C
C ROUTINES:  NONE
C
C NOTE:
C            TRANSCRIBED FROM LINPACK PUBLICATION.  VERSION DATED
C            08/14/78, JACK DONGARRA, ARGONNE NATIONAL LABORATORY
C
C AUTHOR:  H. P. SUMMERS, UNIVERSITY OF STRATHCLYDE
C          JA8.08
C          TEL. 0141-553-4196
C
C DATE:    04/07/95
C
C DATE:    10/07/95                        VERSION 1.1
C UPDATE:  TIM HAMMOND, TESSELLA SUPPORT SERVICES PLC
C           - UNIX PORT
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER  N      , IFAIL
      INTEGER  K      , KB     , KP1    , NM1    , NM2
C-----------------------------------------------------------------------
      REAL*8   T
C-----------------------------------------------------------------------
      REAL*8   DL(N)  , D(N)   , DU(N)  , B(N)
C-----------------------------------------------------------------------
```

XXGUID

```
      SUBROUTINE XXGUID( USERID )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXGUID *********************
C
C  PURPOSE: TO FETCH USER IDENTIFIER FROM UNIX
C
C  CALLING PROGRAM:
C
C  SUBROUTINE:
C
C  OUTPUT: (C*6)   USERID   = USER IDENTIFIER
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C
C UNIX-IDL PORT:
C           WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    22ND APRIL 1996
```

```
C
C VERSION: 1.1                              DATE: 22-04-96
C MODIFIED: WILLIAM OSBORN
C           - FIRST VERSION.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER    USERID*10
      INTEGER PIPEIN
      PARAMETER (PIPEIN=5)
C-----------------------------------------------------------------------
```

XXHKEY

```
      SUBROUTINE XXHKEY( CTEXT  , CKEY , CBREAK , CANS )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXHKEY *********************
C
C  PURPOSE: TO EXTRACT FROM A LINE OF TEXT 'CTEXT' A RESPONSE TO A KEY
C           IN THER FORM OF '<CKEY> = <CANS>'.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (C*(*)) CTEXT   = INPUT TEXT LINE CONTAINING KEY & RESPONSES
C  INPUT : (C*(*)) CKEY    = KEY TEXT
C  INPUT : (C*1  ) CBREAK  = KEY/RESPONSE PAIR SEPERATOR SYMBOL
C
C  OUTPUT: (C*(*)) CANS    = RERSPONSE FOR GIVEN KEY: BLANK IF NOT FOUND
C
C          (I*4)   LENTXT  = LENGTH IN BYTES OF 'CTEXT' STRING
C          (I*4)   LENKEY  = LENGTH IN BYTES OF 'CKEY' STRING
C          (I*4)   LENANS  = LENGTH IN BYTES OF 'CANS' STRING
C          (I*4)   IKEY    = LENGTH IN BYTES OF 'CKEY' IGNORING TRAILING
C                            BLANKS
C          (I*4)   IPOS1   = USED IN IDENTIFYING RELEVANT BYTES IN CTEXT
C          (I*4)   IPOS2   = USED IN IDENTIFYING RELEVANT BYTES IN CTEXT
C          (I*4)   IPOS3   = USED IN IDENTIFYING RELEVANT BYTES IN CTEXT
C          (I*4)   I       = GENERAL USE INDEX
C
C ROUTINES: NONE
C
C NOTES:    THIS ROUTINE EXTRACTS FROM 'CTEXT' A RESPONSE TO A GIVEN KEY
C           IN THER FORM OF '<CKEY> = <CANS>'. E.G. 'FILE = DSN001'
C           WOULD REQUIRE AS INPUT CKEY='FILE' AND WOULD GIVE AS OUTPUT
C           CANS='DSN001'. ALL KEY/RESPONSE PAIRS MUST BE SEPARATED BY
C           THE CHARACTER GIVEN BY 'CBREAK' E.G. A SLASH, AND EACH KEY
C           MUST BE FOLLOWED BY AN EQUALS SIGN. THE NUMBER OF SPACES
C           BETWEEN THE KEY AND THE EQUAL SIGN AND BETWEEN THE RESPONSE
C           AND THE EQUAL SIGN IS NOT IMPORTANT.
C
C           THE BYTE PRECEEDING THE KEY MUST BE A BLANK OR 'CBREAK'
C           CHARACTER UNLESS IT STARTS AT BYTE ONE IN 'CTEXT'.
C
C           IF A KEY DOES NOT EXIST IN 'CTEXT' THEN 'CANS' IS RETURNED
C           BLANK.
C
C           THE KEY IS TAKEN AS 'CKEY' REMOVING ANY TRAILING BLANKS.
C           LEADING BLANKS ARE LEFT IN PLACE AND WILL USED WHEN THE
C           THE SEARCH FOR THE KEY IS MADE:
C
C           I.E.  'DATA   ' AND 'DATA' ARE THE SAME KEY BUT
C                 ' DATA '  AND 'DATA ' ARE DIFFERENT KEYS ALTHOUGH
C                 BOTH WILL GIVE THE SAME RESULTS IF A SPACE EXISTS
C                 BEFORE 'DATA' IN THE INPUT TEXT LINE.
C
C           AN EXAMPLE OF AN INPUT TEXT LINE IS:
C
C           8524.0 A    5 7 /FILMEM = FBBH91BE/   CODE=   V2B DLN1   /
C
C           THIS WOULD GIVE THE FOLLOWING:
C
C           CKEY='FILMEM'  =>  CANS='FBBH91BE'
C           CKEY=' FILMEM' =>  CANS=' '
C           CKEY='CODE'    =>  CANS='V2B DLN1'
C           CKEY=' CODE'   =>  CANS='V2B DLN1'
C           CKEY='OTHER'   =>  CANS=' '
C
C           (IF THE CHARACTER STRING IS SHORTER THAN THE RESPONSE THEN
C            THE RESPONSE IS TRUNCATED ACCORDINGLY.)
C
C           SPACES CAN EXIST IN THE KEY. I.E. CKEY='PLOT A'. BUT CARE
C           SHOULD BE TAKEN WHEN USING PREFIXES ON A COMMON KEY BASE,
C           I.E. 'A PLOT', 'B PLOT'. THIS IS BECAUSE IF A SUBSEQUENT
C           KEY TO BE FOUND IS 'PLOT' THEN  EITHER OF THESE SATISFY
C           THIS CRITERION AS WELL AS 'PLOT' ITSELF.
C
C           AN EXAMPLE OF AN INPUT TEXT LINE IS:
C
C           A FILE=TEST0/B FILE = TEST1/FILE=TEST2/FILE 1=TEST3/FILE 2=/
```

```
C
C           THIS WOULD GIVE THE FOLLOWING:
C
C           CKEY='A FILE'  =>  CANS='TEST0'
C           CKEY='B FILE'  =>  CANS='TEST1'
C           CKEY='FILE'    =>  CANS='TEST0' (WRONG RESPONSE PICKED UP)
C           CKEY='FILE 1'  =>  CANS='TEST3'
C           CKEY='FILE 2'  =>  CANS='TEST4'
C
C           IT IS ALSO POSSIBLE TO IMBED RESPONSES
C
C           AN EXAMPLE OF AN INPUT TEXT LINE IS:
C
C           FILE 1 =  Z1 = 23 / FILE = FILE 1 = 6 /
C
C           THIS WOULD GIVE THE FOLLOWING:
C
C           CKEY='FILE 1'  =>  CANS='Z1 = 23'
C           CKEY=' FILE 1' =>  CANS='6'
C           CKEY='Z1'      =>  CANS='23'
C           CKEY='FILE'    =>  CANS='FILE 1 = 6'
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2520
C
C DATE:    26/04/91
C
C-----------------------------------------------------------------------
      INTEGER    LENTXT      , LENKEY      , LENANS      ,
     &           IPOS1       , IPOS2       , IPOS3       ,
     &           IKEY        , I
C-----------------------------------------------------------------------
      CHARACTER  CTEXT*(*)   , CKEY*(*)    , CBREAK*1    , CANS*(*)
C-----------------------------------------------------------------------
```

XXIDTL

```
       SUBROUTINE XXIDTL( INDEX , N , L )
       IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXIDTL *********************
C
C  PURPOSE:  INVERSE OF FUNCTION I4IDFL. RETURNS THE UNIQUE N AND L
C            QUANTUM NUMBERS WHICH GENERATE THE GIVEN INDEX WHEN PASSED
C            TO I4IDFL.
C
C  CALLING PROGRAM: GENERAL USE.
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  INDEX    = INDEX NUMBER.
C
C  OUTPUT: (I*4)  N       = N QUANTUM NUMBER.
C  OUTPUT: (I*4)  L       = L QUANTUM NUMBER.
C
C          (I*4)  ID      = INDEX RETURNED BY FUNCTION I4IDFL
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4IDFL     ADAS      RETURNS UNIQUE INDEX FROM QUANTUM
C                               NUMBERS N AND L.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    10/09/93
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER   I4IDFL
      INTEGER   INDEX  , N      , L
      INTEGER   ID
C-----------------------------------------------------------------------
```

XXIDTM

```
       SUBROUTINE XXIDTM( INDEX , N , L , M )
       IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXIDTM *********************
C
C  PURPOSE:  INVERSE OF FUNCTION I4IDFM. RETURNS THE UNIQUE N, L AND M
C            QUANTUM NUMBERS WHICH GENERATE THE GIVEN INDEX WHEN PASSED
C            TO I4IDFM.
C
C  CALLING PROGRAM: GENERAL USE.
C
```

```
C  SUBROUTINE:
C
C  INPUT : (I*4)  INDEX   = INDEX NUMBER.
C
C  OUTPUT: (I*4)  N       = N QUANTUM NUMBER.
C  OUTPUT: (I*4)  L       = L QUANTUM NUMBER.
C  OUTPUT: (I*4)  M       = M QUANTUM NUMBER.
C
C          (I*4)  ID      = INDEX RETURNED BY FUNCTION I4IDFM
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4IDFM     ADAS      RETURNS UNIQUE INDEX FROM QUANTUM
C                               NUMBERS N, L AND M.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    10/09/93
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER   I4IDFM
      INTEGER   INDEX  , N      , L      , M
      INTEGER   ID
C-----------------------------------------------------------------------
```

XXIN17

```
      SUBROUTINE XXIN17( IUNIT  , ICLASS , DSNAME , LERROR ,
     &                   NDDEN  , NDTIN  , NDZ1V  ,
     &                   IPRTD  , ISYSD  ,
     &                   IDE    , ITE    , IZE    ,
     &                   DENSR  , TR     , ZIPT   ,
     &                   LSWIT  , EIA    ,
     &                   AIPT
     &                 )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXIN17 *********************
C
C  PURPOSE: TO OPEN AND ACQUIRE DATA FROM THE FOLLOWING MASTER CONDENSED
C           COLLISIONAL-DIELECTRONIC FILES:
C
C             1. RECOMBINATION COEFFICIENTS
C             2. IONISATION    COEFFICIENTS
C             3. CHARGE-EXCHANGE RECOMBINATION COEFFICIENTS
C             4. METASTABLE CROSS-COUPLING COEFFICIENTS
C             5. PARENT METASTABLE CROSS-COUPLING COEFFICIENTS
C             6. RECOMBINATION-BREMSSTRAHLUNG  POWER COEFFICIENTS
C             7. CHARGE-EXCHANGE RECOMBINATION POWER COEFFICIENTS
C
C             (NOTE: SPECIFIC AND TOTAL LOW LINE POWER COEFFICIENTS
C                    SHOULD BE READ USING 'XXIN80'.
C                    IF  ONLY STANDARD FILES ARE TO BE READ BY  THE
C                    PROGRAM USE 'XXINST'.)
C
C  CALLING PROGRAM: GENERAL USE
C
C  DATA:
C           THE  SOURCE  DATA IS CONTAINED AS  MEMBERS  OF  PARTITIONED
C           DATA SETS AS FOLLOWS:
C
C             1. JETUID.ACD<YR>.DATA
C             2. JETUID.SCD<YR>.DATA
C             3. JETUID.CCD<YR>.DATA
C             4. JETUID.QCD<YR>.DATA
C             5. JETUID.XCD<YR>.DATA
C             6. JETUID.PRB<YR>.DATA
C             7. JETUID.PRC<YR>.DATA
C
C           WHERE <YR> DENOTES TWO INTEGERS FOR THE YEAR SELECTED.
C           IF <YR> IS BLANK THEN THE CURRENT RECOMMENDED DATA SETS ARE
C           USED
C
C           THE MEMBERS OF THE PARTITIONED DATA SETS ARE EITHER:
C           1)  <SE><I><J>  FOR PARTIAL MASTER CONDENSED FILES, OR
C           2)  <SE>        FOR STANDARD MASTER CONDENSED FILES
C
C           WHERE: <SE> IS THE ONE OR TWO LETTER ION SEQUENCE CODE.
C                  <I>  IS A  SINGLE  INTEGER  REPRESENTING  THE  PARENT
C                       INDEX OR METASTABLE INDEX DEPENDING ON THE DATA
C                       SET CLASS AND PRODUCTION BY BUNDLE-NS OR
C                       LOW-LEVEL+PROJECTION MODELS
C                  <J>  IS A SINGLE INTEGER REPRESENTING THE SPIN SYSTEM
C                       INDEX, METASTABLE INDEX OR PARENT INDEX
C                       DEPENDING ON THE DATA SET CLASS AND PRODUCTION
C                       BY BUNDLE-NS OR LOW-LEVEL+PROJECTION MODELS
C
C                  E.G. PARTIAL  FILES: 'C12' OR 'HE21'
C                       STANDARD FILES: 'C'   OR 'HE'
C
```

```
C           THE  'PARTIAL'  AND  'STANDARD'  MASTER CONDENSED FILES ARE
C           IDENTICAL IN FORM,   EXCEPT  THAT  THREE  ADDITIONAL  LINES
C           ARE INCLUDED  AT  THE  BEGINNING OF  THE  'PARTIAL'  MASTER
C           FILES.   THE  FIRST  OF THESE LINES CONTAINS  A ROW OF  '='
C           SIGNS,  THE SECOND A PARENT/SPIN (OR EQUIVALENTS) PARAMETER
C           LIST,   AND  THE
C           THIRD A ROW OF "-" SIGNS. THIS DIFFERENCE IS USED TO IDENT-
C           IFY WHICH FILE TYPE IS BEING READ.
C
C           THE CHARACTER STRING SEPARATING THE INPUT DATA FOR EACH
C           VALUE OF Z1 IN THE FILE WILL GIVE:
C
C           PARTIAL & STANDARD: THE Z1 VALUE (Z1=) AND DATE (DATE:).
C                (OLDER DATA SETS MAY HAVE 'Z =' INSTEAD OF 'Z1=' HERE)
C           PARTIAL FILES ONLY: THE PARENT (IPRT=) & SPIN SYSTEM (ISYS=)
C                OR EQUIVALENTS (IGRD=) & (IGRD=, JGRD= AND JPRT=)
C                AS FOLLOW:-
C
C               ICLASS     INDI     INDJ
C               ------     ----     ----
C                 1        IPRT     IGRD (OR ISYS)
C                 2        IPRT     IGRD (OR ISYS)
C                 3        IPRT     IGRD (OR ISYS)
C                 4        IPRT     IGRD (OR ISYS)
C                 5        IPRT     IGRD (OR ISYS)
C                 6        IGRD     JGRD
C                 7        IPRT     JPRT
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH INPUT DATA SET ALLOCATED
C  INPUT : (I*4)  ICLASS  = UNIT TO WHICH INPUT DATA SET ALLOCATED
C  INPUT : (C*(*))DSNAME  = INPUT MASTER CONDENSED FILE DATA SET NAME
C  OUTPUT: (L*4)  LERROR  = .TRUE.  => ERROR DETECTED IN READING FILE
C                         = .FALSE. => NO ERROR DETECTED IN FILE
C
C  INPUT : (I*4)  NDDEN   = MAX. NUMBER OF REDUCED DENSITIES ALLOWED IN
C                           MASTER CONDENSED FILE FOR A GIVEN SEQUENCE
C  INPUT : (I*4)  NDTIN   = MAX. NO. OF REDUCED TEMPERATURES ALLOWED IN
C                           MASTER CONDENSED FILE FOR A GIVEN SEQUENCE
C  INPUT : (I*4)  NDZ1V   = MAX. NUMBER OF CHARGE STATES ALLOWED IN
C                           MASTER CONDENSED FILE FOR A GIVEN SEQUENCE
C
C  INPUT : (I*4)  IPRTD   = INPUT PARTIAL MASTER CONDENSED FILE:
C                           PARENT INDEX SPECIFIED IN DATA SET NAME.
C  INPUT : (I*4)  ISYSD   = INPUT PARTIAL MASTER CONDENSED FILE: SPIN-
C                           SYSTEM INDEX SPECIFIED IN DATA SET NAME.
C
C  OUTPUT: (I*4)  IDE     = NUMBER OF REDUCED DENSITIES READ FROM INPUT
C                           MASTER CONDENSED FOR A GIVEN SEQUENCE
C  OUTPUT: (I*4)  ITE     = NO. OF REDUCED TEMPERATURES READ FROM INPUT
C                           MASTER CONDENSED FOR A GIVEN SEQUENCE
C  OUTPUT: (I*4)  IZE     = NO. OF CHARGE STATES GIVEN IN THE INPUT
C                           MASTER CONDENSED FOR A GIVEN SEQUENCE
C
C  OUTPUT: (R*8)  DENSR() = SET OF 'IDE' INPUT REDUCED DENSITIES (CM-3/
C                           Z1**7) READ FROM CONDENSED MASTER FILE.
C  OUTPUT: (R*8)  TR()    = SET OF 'ITE'  INPUT  REDUCED  TEMPERATURES
C                           (K/Z1**2) READ FROM CONDENSED MASTER FILE.
C  OUTPUT: (R*8)  ZIPT()  = SET OF 'IZE' INPUT CHARGE STATES READ FROM
C                           CONDENSED MASTER FILE.
C                           (CHARGE STATE = RECOMBINING ION CHARGE)
C
C  OUTPUT: (L*4)  LSWIT   = .TRUE.  => IONISATION POTENTIALS
C                                      INCLUDED IN INPUT MASTER FILE.
C                           .FALSE. => IONISATION POTENTIALS
C                                      NOT INCLUDED IN INPUT MASTER FILE
C  OUTPUT: (R*8)  EIA()   = IONISATION POTENTIALS: ()=ION CHARGE
C                           UNITS: WAVE NUMBERS (CM-1)
C                           (= 0.0 IF NOT SET)
C
C  OUTPUT: (R*8)  AIPT(,,)= CONDENSED MASTER FILE DATA. COLL-DIEL COEFF.
C                           1ST DIMENSION: REDUCED DENSITY ('DENSR()')
C                           2ND DIMENSION: REDUCED TEMPERATURE ('TR()')
C                           3RD DIMENSION: CHARGE STATE ('ZIPT()')
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  IPRT    = INPUT PARTIAL MASTER CONDENSED FILE:
C                           PARENT INDEX READ FROM INPUT FILE.
C          (I*4)  ISYS    = INPUT PARTIAL MASTER CONDENSED FILE:
C                           SPIN-SYSTEM INDEX READ FROM INPUT FILE.
C          (I*4)  IPOT    = NUMBER OF IONISATION POTENTIAL VALUES
C                           PRESENT IN THE INPUT FILE.
C          (I*4)  IZ1     = CHARGE STATE READ FROM THE LINE PRECEEDING
C                           AN INPUT BLOCK FROM THE FILE.
C                           (= RECOMBINING ION CHARGE)
C          (I*4)  IBGN    = FIRST BYTE OF INTEREST IN CHARACTER 'STRING'
C          (I*4)  IEND    = LAST  BYTE OF INTEREST IN CHARACTER 'STRING'
C          (I*4)  ID      = ARRAY SUBSCRIPT USED FOR DENSITY VALUES
C          (I*4)  IT      = ARRAY SUBSCRIPT USED FOR TEMPERATURE VALUES
C          (I*4)  IZ      = ARRAY SUBSCRIPT USED FOR ION-CHARGE VALUES
C          (I*4)  I       = GENERAL USE
C
C          (L*4)  LPART   = .TRUE.  => REQUESTED INPUT FILE: PARTIAL
C                           = .FALSE. => REQUESTED INPUT FILE: STANDARD
C
C          (C*5)  CPOT    = 'IPOT'
```

```
C          (C*5)  CHINDI  = 'IPRT= ' OR 'IGRD= ' DEPENDING ON ICLASS
C          (C*5)  CHINDJ  = 'IGRD= ', 'JGRD= ' OR 'JPRT= ' DEPENDING
C                           ON ICLASS
C          (C*80) STRING  = STRING INTO WHICH 1ST LINE OF INPUT FILE IS
C                           READ TO ENABLE ITS FORMAT TO BE ESTABLISHED.
C
C NOTE:
C          STREAM HANDLING:
C             STREAM 'IUNIT' IS USED FOR READING CONDENSED MASTER FILES
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          XXREIA     ADAS      READ IN UNKNOWN NUMBER OF 'EIA' VALUES
C                               IF PRESENT.
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    22/08/90
C
C UPDATE:  05/03/91 - PE BRIDEN - ADAS91: REMOVED OPENING OF DATA SET
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C UPDATE:  11/08/93 - HP SUMMERS - CHANGED TO ACCEPT EXTRA DATA CLASSES
C                                  AND USE OF IGRD, JGRD, IPRT, JPRT AS
C                                  ALTERNATIVES TO IPRT AND ISYS.
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER    I4UNIT
      INTEGER    IUNIT       , NDDEN     , NDTIN     , NDZ1V    ,
     &           IPRTD       , ICLASS    ,
     &           ISYSD       ,
     &                         IDE       , ITE       , IZE
      INTEGER    IPRT        , ISYS      ,
     &           IPOT        , IZ1       , IBGN      , IEND     ,
     &           ID          , IT        , IZ        , I
C-----------------------------------------------------------------------
      CHARACTER  CPOT*5      , CHINDI*5  , CHINDJ*5  ,
     &           DSNAME*(*)  , STRING*80
C-----------------------------------------------------------------------
      LOGICAL    LERROR      , LSWIT       , LPART
C-----------------------------------------------------------------------
      REAL*8     DENSR(NDDEN), TR(NDTIN)   , ZIPT(NDZ1V)
      REAL*8     EIA(250)
      REAL*8     AIPT(NDDEN,NDTIN,NDZ1V)
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
```

XXIN80

```
      SUBROUTINE XXIN80( IUNIT  , DSNAME , LERROR ,
     &                   NDDEN  , NDTIN  , NDZ1V  , NDMET ,
     &                   IDE    , ITE    , IZE    ,
     &                   DENSR  , TR     , ZIPT   ,
     &                   IME    , IMETR  , CSTRGA ,
     &                   NPRNT  , IPRNT  , IPSYS  ,
     &                   LSWIT  , EIA    ,
     &                   AIPT
     &                 )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXIN80 *********************
C
C  PURPOSE: TO OPEN AND ACQUIRE DATA FROM THE FOLLOWING MASTER CONDENSED
C           COLLISIONAL-DIELECTRONIC FILES:
C
C             8. TOTAL    LINE POWER COEFFICIENTS
C             9. SPECIFIC LINE POWER COEFFICIENTS
C
C           AND TO OPEN AND ACQUIRE DATA FROM THE FOLLOWING FILE:
C
C            10. METASTABLE POPULATION DATA
C
C             (NOTE: OTHER MASTER  CONDENSED  COLL.-DIEL.  COEFFICIENTS
C                    SHOULD BE READ USING 'XXIN17'.
C                    IF  ONLY STANDARD FILES ARE  TO  BE  READ  BY  THE
C                    PROGRAM USE 'XXINST'.)
```

```
C
C  CALLING PROGRAM: GENERAL USE
C
C  DATA:
C           THE  SOURCE  DATA IS CONTAINED AS  MEMBERS  OF  PARTITIONED
C           DATA SETS AS FOLLOWS:
C
C             8. JETUID.PLT<YR>.DATA
C             9. JETUID.PLS<YR>.DATA
C            10. JETUID.MET<YR>.DATA
C
C           WHERE <YR> DENOTES TWO INTEGERS FOR THE YEAR SELECTED.
C           IF <YR> IS BLANK THEN THE CURRENT RECOMMENDED DATA SETS ARE
C           USED
C
C           THE MEMBERS OF THE PARTITIONED DATA SETS ARE EITHER:
C           1)  <SE><#><#>  FOR PARTIAL MASTER CONDENSED FILES, OR
C           2)  <SE>        FOR STANDARD MASTER CONDENSED FILES AND
C                           METASTABLE POPULATION FILE
C
C           WHERE: <SE> IS THE ONE OR TWO LETTER ION SEQUENCE CODE.
C                       NOTE: FOR THE BARE NUCLEUS <SE> = '@'
C                  <#>  IS THE SINGLE CHARACTER '#'
C
C
C                  E.G. PARTIAL  MASTER CONDENSED FILES:'@##' OR 'HE##'
C                       STANDARD AND METASTABLE FILES  : '@'  OR 'HE'
C
C           THE  'PARTIAL'  AND  'STANDARD'  MASTER CONDENSED FILES ARE
C           IDENTICAL IN FORM  EXCEPT:
C           A) EIGHT  ADDITIONAL  LINES ARE INCLUDED  AT THE  BEGINNING
C              OF  THE  'PARTIAL'  MASTER  FILES.   THE FIRST  OF THESE
C              LINES CONTAINS  A ROW OF  '=' SIGNS,  THE NEXT SIX LINES
C              CONTAIN METASTABLE LEVEL  INFORMATION, SUCH AS NUMBER OF
C              LEVELS, PARENT/SPIN INDEXES ETC. THE EIGHTH LINE CONTAINS
C              A ROW OF "-" SIGNS.  THIS DIFFERENCE IS USED TO IDENTIFY
C              WHICH FILE TYPE IS BEING READ.
C           B) THE PARTIAL FILES INCLUDE THE POPULATIONS  FOR  EACH  OF
C              THE METASTABLE LEVELS, WHEREAS THE STANDARD FILES CONTAIN
C              A SINGLE SET OF COMBINED POPULATIONS.
C
C          THE METASTABLE POPULATION FILES HAVE THE SAME FORMAT AS THE
C          PARTIAL MASTER CONDENSED FILES
C
C          THE CHARACTER STRING SEPARATING THE INPUT DATA FOR EACH
C          CHARGE STATE IN THE FILE WILL GIVE:
C
C          THE CHARGE STATE VALUE (Z1=) AND DATE (DATE:).
C                (OLDER DATA SETS MAY HAVE 'Z =' INSTEAD OF 'Z1=' HERE)
C          (CHARGE STATE Z1 = ION CHARGE + 1 = RECOMBINING ION CHARGE)
C
C          UNDER EACH OF THESE LINES THE COEFFTS/POPULATIONS ARE LISTED
C          -IN THE CASE OF THE METASTABLE/PARTIAL FILES VALUES FOR EACH
C          OF THE METASTABLE LEVELS ARE LISTED EACH BEING  PRECEDED  BY
C          A LINE GIVING THE METASTABLE INDEX BETWEEN TWO "/".
C          E.G.  /2/ => METASTABLE LEVEL TWO.
C
C          DATA FOR INDIVIDUAL PARENT/SPIN SYSTEMS ARE LISTED IN THE
C          MEATASTABLE FILES.
C
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH INPUT DATA SET ALLOCATED
C  INPUT : (C*(*))DSNAME  = INPUT MASTER CONDENSED FILE DATA SET NAME
C  OUTPUT: (L*4)  LERROR  = .TRUE.  => ERROR DETECTED IN READING FILE
C                         = .FALSE. => NO ERROR DETECTED IN FILE
C
C  INPUT : (I*4)  NDDEN   = MAX. NUMBER OF REDUCED DENSITIES ALLOWED IN
C                           MASTER CONDENSED/METASTABLE FILE FOR A GIVEN
C                           SEQUENCE.
C  INPUT : (I*4)  NDTIN   = MAX. NO. OF REDUCED TEMPERATURES ALLOWED IN
C                           MASTER CONDENSED/METASTABLE FILE FOR A GIVEN
C                           SEQUENCE.
C  INPUT : (I*4)  NDZ1V   = MAX. NUMBER OF CHARGE STATES ALLOWED IN
C                           MASTER CONDENSED/METASTABLE FILE FOR A GIVEN
C                           SEQUENCE.
C  INPUT : (I*4)  NDMET   = MAX.  NUMBER OF METASTABLE STATES ALLOWED IN
C                           MASTER CONDENSED/METASTABLE FILE FOR A GIVEN
C                           SEQUENCE.
C                           NOT USED FOR STANDARD MASTER CONDENSED FILES
C                           (SET EQUAL TO 1 IN THIS CASE).
C
C  OUTPUT: (I*4)  IDE     = NUMBER OF REDUCED DENSITIES READ FROM INPUT
C                           MASTER CONDENSED/METASTABLE FILE FOR A GIVEN
C                           SEQUENCE.
C  OUTPUT: (I*4)  ITE     = NO. OF REDUCED TEMPERATURES READ FROM INPUT
C                           MASTER CONDENSED/METASTABLE FILE FOR A GIVEN
C                           SEQUENCE.
C  OUTPUT: (I*4)  IZE     = NO. OF CHARGE STATES GIVEN IN THE INPUT
C                           MASTER CONDENSED/METASTABLE FILE FOR A GIVEN
C                           SEQUENCE.
C
C  OUTPUT: (R*8)  DENSR() = SET OF 'IDE' INPUT REDUCED DENSITIES (CM-3/
C                           Z1**7) READ FROM CONDENSED MASTER/METASTABLE
C                           FILE.
C  OUTPUT: (R*8)  TR()    = SET OF 'ITE'  INPUT  REDUCED  TEMPERATURES
C                           (K/Z1**2) READ FROM CONDENSED MASTER/
```

```
C                              METASTABLE FILE.
C  OUTPUT: (R*8)  ZIPT()  = SET OF 'IZE' INPUT CHARGE STATES READ FROM
C                           CONDENSED MASTER/METASTABLE FILE.
C                           (CHARGE STATE = ION CHARGE + 1 = RECOMBINING
C                            ION CHARGE)
C
C  OUTPUT: (I*4)  IME     = NO. OF METASTABLE LEVELS CONTAINED IN THE
C                           INPUT MASTER CONDENSED/METASTABLE FILE.
C                           EQUALS 1 FOR STANDARD MASTER CONDENSED FILES
C  OUTPUT: (I*4)  IMETR() =THE ORIGINAL COPDAT INDEX FOR EACH METASTABLE
C                           LEVEL. DIMENSION: METASTABLE LEVEL INDEX.
C                           NOT USED FOR STANDARD MASTER CONDENSED FILES
C  OUTPUT: (C*12) CSTRGA()=THE DESIGNATION OF EACH METASTABLE LEVEL.
C                           DIMENSION: METASTABLE LEVEL INDEX.
C                           NOT USED FOR STANDARD MASTER CONDENSED FILES
C
C  OUTPUT: (I*4)  NPRNT   = NUMBER OF PARENTS CONTAINED IN THE INPUT
C                           MASTER CONDENSED/METASTABLE FILE.
C                           NOT USED FOR STANDARD MASTER CONDENSED FILES
C                           (NOTE: THE NUMBER OF PARENTS CANNOT EXCEED
C                                  THE NUMBER OF METASTABLE LEVELS)
C  OUTPUT: (I*4)  IPRNT() = THE PARENT INDEX FOR INPUT PARENT.
C                           DIMENSION: PARENT/(METASTABLE LEVEL) INDEX.
C                           NOT USED FOR STANDARD MASTER CONDENSED FILES
C  OUTPUT: (I*4)  IPSYS() = THE SPIN SYSTEM REFERENCE FOR EACH INPUT
C                           PARENT.
C                           DIMENSION: PARENT/(METASTABLE LEVEL) INDEX.
C                           NOT USED FOR STANDARD MASTER CONDENSED FILES
C
C  OUTPUT: (L*4)  LSWIT   = .TRUE.  => IONISATION POTENTIALS
C                                      INCLUDED IN INPUT MASTER FILE.
C                           .FALSE. => IONISATION POTENTIALS
C                                      NOT INCLUDED IN INPUT MASTER FILE
C  OUTPUT: (R*8)  EIA()   = IONISATION POTENTIALS: ()=ION CHARGE
C                           UNITS: WAVE NUMBERS (CM-1)
C                           (= 0.0 IF NOT SET)
C
C  OUTPUT: (R*8) AIPT(,,,)= OPTION 6: TOTAL    LINE POWER COEFFICIENTS
C                           OPTION 7: SPECIFIC LINE POWER COEFFICIENTS
C                           OPTION 8: METASTABLE STATE POPULATIONS
C                           1ST DIMENSION: ELECTRON DENSITY INDEX
C                                          ('DENSR()')
C                           2ND DIMENSION: ELECTRON TEMPERATURE INDEX
C                                          ('TR()')
C                           3RD DIMENSION: CHARGE STATE INDEX
C                                          ('ZIPT()')
C                           4TH DIMENSION: METASTABLE STATE INDEX
C                           (OPTIONS 6 & 7 STANDARD FILES ALWAYS = 1)
C
C          (C*1)  CBLNK   = PARAMETER = ' '
C          (C*1)  CEQUAL  = PARAMETER = '='
C          (C*1)  CSTAR   = PARAMETER = '*'
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  IPOT    = NUMBER OF IONISATION POTENTIAL VALUES
C                           PRESENT IN THE INPUT FILE.
C          (I*4)  IZ1     = CHARGE STATE READ FROM THE LINE PRECEEDING
C                           AN INPUT BLOCK FROM THE FILE.
C                           (= ION CHARGE + 1 = RECOMBINING ION CHARGE)
C          (I*4)  IMET    = METASTABLE STATE OF CURRENT DATA BLOCK BEING
C                           READ.
C          (I*4)  IBGN    = FIRST BYTE OF INTEREST IN CHARACTER 'STRING'
C          (I*4)  IEND    = LAST  BYTE OF INTEREST IN CHARACTER 'STRING'
C          (I*4)  ID      = ARRAY SUBSCRIPT USED FOR DENSITY INDEXES
C          (I*4)  IT      = ARRAY SUBSCRIPT USED FOR TEMPERATURE INDEXS
C          (I*4)  IZ      = ARRAY SUBSCRIPT USED FOR ION-CHARGE INDEXES
C          (I*4)  IM      = ARRAY SUBSCRIPT USED FOR METASTABLE INDEXES
C          (I*4)  I       = GENERAL USE
C
C          (L*4)  LPART   = .TRUE.  => REQUESTED INPUT FILE: PARTIAL
C                                                  OR METASTABLE POPULATION.
C                           .FALSE. => REQUESTED INPUT FILE: STANDARD
C
C          (C*1)  C1      = GENERAL USE 1-BYTE CHARACTER STRING.
C                           (STORES METASTABLE STATE ORDER INDEX).
C          (C*5)  CPOT    = 'IPOT'
C
C          (C*80) STRING()= STRINGS  INTO WHICH LINES OF INPUT FILE ARE
C                           READ TO ENABLE ITS FORMAT TO BE ESTABLISHED
C                           AND CONTENTS READ.
C
C
C NOTE:
C          STREAM HANDLING:
C             STREAM 'IUNIT' IS USED FOR READING CONDENSED MASTER FILES
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          XXREIA     ADAS      READ IN UNKNOWN NUMBER OF 'EIA' VALUES
C                               IF PRESENT.
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
```

```
C DATE:    05/03/91 (DIFFERENT FROM ADAS90 VERSION - REMOVED DSN OPEN)
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C UPDATE:  11/08/93 - HP SUMMERS - RENAMED TO XXIN80 FROM XXIN68
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE (NO CHANGES)
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      CHARACTER  CBLNK        , CEQUAL       , CSTAR
C-----------------------------------------------------------------------
      PARAMETER( CBLNK = ' '  , CEQUAL = '=' , CSTAR = '*'  )
C-----------------------------------------------------------------------
      INTEGER    I4UNIT
      INTEGER    IUNIT        ,
     &           NDDEN        , NDTIN     , NDZ1V     , NDMET   ,
     &           IDE          , ITE       , IZE       , IME     , NPRNT
      INTEGER    IPOT         , IZ1       , IMET      ,
     &           IBGN         , IEND      ,
     &           ID           , IT        , IZ        , IM      , I
C-----------------------------------------------------------------------
      CHARACTER  C1*1         , CPOT*5
      CHARACTER  DSNAME*(*)
C-----------------------------------------------------------------------
      LOGICAL    LERROR       , LSWIT     , LPART
C-----------------------------------------------------------------------
      INTEGER    IMETR(NDMET)      , IPRNT(NDMET)      , IPSYS(NDMET)
C-----------------------------------------------------------------------
      REAL*8     DENSR(NDDEN)      , TR(NDTIN)         , ZIPT(NDZ1V)
      REAL*8     EIA(250)
      REAL*8     AIPT(NDDEN,NDTIN,NDZ1V,NDMET)
C-----------------------------------------------------------------------
      CHARACTER  CSTRGA(NDMET)*12
      CHARACTER  STRING(5)*80
C-----------------------------------------------------------------------
```

XXINDX

```
      SUBROUTINE XXINDX( N , ARR , INDX )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXINDX *********************
C
C  PURPOSE: INDEXES AN ARRAY 'ARR' OF LENGTH 'N'. IT OUTPUTS THE ARRAY
C           'INDX()' SUCH THAT 'ARR(INDX(J))' IS  IN  ASCENDING  ORDER
C           FOR J=1,2,...,N.   THE INPUT QUANTITIES 'N' AND 'ARR'  ARE
C           NOT CHANGED.
C
C  REFERENCE:  NUMERICAL RECIPES: The Art of Scientific Computing
C              (FORTRAN Version).
C              W.H.Press, B.P.Flannery, S.A.Teukolsky & W.T.Vetterling.
C              (Cambridge University Press, Cambridge). 1989. p.233
C              ISBN 0 521 38330 7
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT :    (I*4)  N       = INPUT ARRAY 'ARR()' LENGTH
C  INPUT :    (R*8)  ARR()   = ARRAY TO BE INDEXED
C  OUTPUT:    (I*4)  INDX()  = ASCENDING ORDER INDEX
C
C             (R*8)  INDXT   = TEMPORARY STORAGE FOR 'INDX' ELEMENT
C             (I*4)  L       = "HIRING" PHASE INDEX
C             (I*4)  IR      = "RETIREMENT-AND-PROMOTION"  PHASE INDEX
C             (I*4)  I       = GENERAL USE
C             (I*4)  J       = GENERAL USE
C
C             (R*8)  Q       = TEMPORARY STORAGE FOR 'ARR' ELEMENT
C
C ROUTINES:  NONE
C
C NOTE:      USES THE HEAPSORT METHOD.
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    09/10/90
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER N        , INDXT     , L        , IR       , I       , J
      INTEGER INDX(N)
```

```
C-----------------------------------------------------------------------
      REAL*8  Q
      REAL*8  ARR(N)
C-----------------------------------------------------------------------
```

XXINST

```
      SUBROUTINE XXINST( IUNIT  , DSNAME , LERROR ,
     &                   NDDEN  , NDTIN  , NDZ1V  ,
     &                   IDE    , ITE    , IZE    ,
     &                   DENSR  , TR     , ZIPT   ,
     &                   LSWIT  , EIA    ,
     &                   AIPT
     &                 )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXINST *********************
C
C  PURPOSE: TO FETCH DATA FROM THE FOLLOWING STANDARD MASTER CONDENSED
C           COLLISIONAL-DIELECTRONIC FILES:
C
C             1. RECOMBINATION COEFFICIENTS
C             2. IONISATION    COEFFICIENTS
C             3. CHARGE-EXCHANGE RECOMBINATION COEFFICIENTS
C             4. RECOMBINATION-BREMSSTRAHLUNG  POWER COEFFICIENTS
C             5. CHARGE-EXCHANGE RECOMBINATION POWER COEFFICIENTS
C             6. TOTAL    LINE POWER COEFFICIENTS
C             7. SPECIFIC LINE POWER COEFFICIENTS
C
C             (NOTE: PARTIAL MASTER CONDENSED FILES SHOULD BE READ USING
C                    'XXIN15' & 'XXIN68'.
C                    IF  PARTIAL AND STANDARD FILES ARE BOTH TO BE READ
C                    USE 'XXIN15' AND 'XXIN68')
C
C  CALLING PROGRAM: GENERAL USE
C
C  DATA:
C           THE  SOURCE  DATA IS CONTAINED AS  MEMBERS  OF  PARTITIONED
C           DATA SETS AS FOLLOWS:
C
C             1. JETUID.ACD<YR>.DATA
C             2. JETUID.SCD<YR>.DATA
C             3. JETUID.CCD<YR>.DATA
C             4. JETUID.PRB<YR>.DATA
C             5. JETUID.PRC<YR>.DATA
C             5. JETUID.PLT<YR>.DATA
C             5. JETUID.PLS<YR>.DATA
C
C           WHERE <YR> DENOTES TWO INTEGERS FOR THE YEAR SELECTED.
C           IF <YR> IS BLANK THEN THE CURRENT RECOMMENDED DATA SETS ARE
C           USED
C
C           THE MEMBERS OF THE PARTITIONED DATA SETS MUST BE STANDARDC           MASTER
CONDENSED FILES AND OF THE FORM <SE>
C
C           WHERE: <SE> IS THE ONE OR TWO LETTER ION SEQUENCE CODE.
C
C                  E.G.  'C'   OR 'HE'  ('@' = BARE-NUCLEUS)
C
C                  PARTIAL  FILES WHICH HAVE MEMBERS NAMES WITH EITHER
C                  TWO DIGITS OR A DOUBLE HASH  AFTER THEM  CANNOT  BE
C                  READ USING THIS ROUTINE (SEE 'XXIN15' & 'XXIN68')
C
C           THE  'PARTIAL'  AND  'STANDARD'  MASTER CONDENSED FILES  ARE
C           SIMILAR IN FORM. A MAJOR DIFFERENCE IS THAT ADDITIONAL LINES
C           ARE  INCLUDED  AT  THE  BEGINNING OF  THE  'PARTIAL'  MASTER
C           FILES.    THE  FIRST OF THESE LINES CONTAINING A ROW OF  '='
C           SIGNS.  THIS DIFFERENCE IS USED TO MAKE SURE THAT A STANDARD
C           MASTER FILE IS BEING READ.
C
C           THE CHARACTER STRING SEPARATING THE INPUT DATA FOR EACH
C           VALUE OF Z1 IN THE FILE WILL GIVE:
C
C                       THE Z1 VALUE (Z1=) AND DATE (DATE:).
C              (OLDER DATA SETS MAY HAVE 'Z =' INSTEAD OF 'Z1=' HERE)
C
C           (NOTE: Z1 = ION CHARGE + 1 = RECOMBINING ION CHARGE)
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH INPUT DATA SET ALLOCATED
C  INPUT : (C*(*))DSNAME  = INPUT MASTER CONDENSED FILE DATA SET NAME
C  OUTPUT: (L*4)  LERROR  = .TRUE.  => ERROR DETECTED IN READING FILE
C                         = .FALSE. => NO ERROR DETECTED IN FILE
C
C  INPUT : (I*4)  NDDEN   = MAX. NUMBER OF REDUCED DENSITIES ALLOWED IN
C                           MASTER CONDENSED FILE FOR A GIVEN SEQUENCE
C  INPUT : (I*4)  NDTIN   = MAX. NO. OF REDUCED TEMPERATURES ALLOWED IN
C                           MASTER CONDENSED FILE FOR A GIVEN SEQUENCE
C  INPUT : (I*4)  NDZ1V   = MAX. NUMBER OF CHARGE STATES ALLOWED IN
C                           MASTER CONDENSED FILE FOR A GIVEN SEQUENCE
C
C  OUTPUT: (I*4)  IDE     = NUMBER OF REDUCED DENSITIES READ FROM INPUT
C                           MASTER CONDENSED FOR A GIVEN SEQUENCE
```

```
C  OUTPUT: (I*4)  ITE     = NO. OF REDUCED TEMPERATURES READ FROM INPUT
C                           MASTER CONDENSED FOR A GIVEN SEQUENCE
C  OUTPUT: (I*4)  IZE     = NO. OF CHARGE STATES GIVEN IN THE INPUT
C                           MASTER CONDENSED FOR A GIVEN SEQUENCE
C
C  OUTPUT: (R*8)  DENSR() = SET OF 'IDE' INPUT REDUCED DENSITIES (CM-3/
C                           Z1**7) READ FROM CONDENSED MASTER FILE.
C  OUTPUT: (R*8)  TR()    = SET OF 'ITE'  INPUT  REDUCED  TEMPERATURES
C                           (K/Z1**7) READ FROM CONDENSED MASTER FILE.
C  OUTPUT: (R*8)  ZIPT()  = SET OF 'IZE' INPUT CHARGE STATES READ FROM
C                           CONDENSED MASTER/METASTABLE FILE.
C                           (CHARGE STATE = ION CHARGE + 1 = RECOMBINING
C
C  OUTPUT: (L*4)  LSWIT   = .TRUE.  => IONISATION RATE COEFFICIENTS
C                                      INCLUDED IN INPUT MASTER FILE.
C                           .FALSE. => IONISATION RATE COEFFICIENTS
C                                      NOT INCLUDED IN INPUT MASTER FILE
C  OUTPUT: (R*8)  EIA()   = IONISATION RATE COEFFICIENTS: ()=ION CHARGE
C                           UNITS: WAVE NUMBERS (CM-1)
C                           (= 0.0 IF NOT SET )
C
C  OUTPUT: (R*8)  AIPT(,,)= CONDENSED MASTER FILE DATA. COLL-DIEL COEFF.
C                           1ST DIMENSION: REDUCED DENSITY ('DENSR()')
C                           2ND DIMENSION: REDUCED TEMPERATURE ('TR()')
C                           3RD DIMENSION: CHARGE STATE ('ZIPT()')
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  IPOT    = NUMBER OF IONISATION RATE COEFF. VALUES
C                           PRESENT IN THE INPUT FILE.
C          (I*4)  IZ1     = CHARGE STATE READ FROM THE LINE PRECEEDING
C                           AN INPUT BLOCK FROM THE FILE.
C                           (= ION CHARGE + 1 = RECOMBINING ION CHARGE)
C          (I*4)  IBGN    = FIRST BYTE OF INTEREST IN CHARACTER 'STRING'
C          (I*4)  IEND    = LAST  BYTE OF INTEREST IN CHARACTER 'STRING'
C          (I*4)  ID      = ARRAY SUBSCRIPT USED FOR DENSITY VALUES
C          (I*4)  IT      = ARRAY SUBSCRIPT USED FOR TEMPERATURE VALUES
C          (I*4)  IZ      = ARRAY SUBSCRIPT USED FOR ION-CHARGE VALUES
C          (I*4)  I       = GENERAL USE
C
C          (C*5)  CPOT    = 'IPOT'
C          (C*80) STRING  = STRING INTO WHICH 1ST LINE OF INPUT FILE IS
C                           READ TO ENABLE ITS FORMAT TO BE ESTABLISHED.
C
C NOTE:
C          STREAM HANDLING:
C             STREAM 'IUNIT' IS USED FOR READING CONDENSED MASTER FILES
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          XXREIA     ADAS      READ IN UNKNOWN NUMBER OF 'EIA' VALUES
C                               IF PRESENT.
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    22/08/90
C
C DATE:    05/03/90 - PE BRIDEN - ADAS91: OPENING OF DATA SET REMOVED.
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER    I4UNIT
      INTEGER    IUNIT       , NDDEN     , NDTIN     , NDZ1V    ,
     &                         IDE       , ITE       , IZE
      INTEGER    IPOT        , IBGN      , IEND      , IZ1      ,
     &           I           , ID        , IT        , IZ
C-----------------------------------------------------------------------
      CHARACTER  DSNAME*(*)
      CHARACTER  CPOT*5      , STRING*80
C-----------------------------------------------------------------------
      LOGICAL    LERROR      , LSWIT
C-----------------------------------------------------------------------
      REAL*8     DENSR(NDDEN), TR(NDTIN) , ZIPT(NDZ1V)
      REAL*8     EIA(50)
      REAL*8     AIPT(NDDEN,NDTIN,NDZ1V)
C-----------------------------------------------------------------------
```

XXISRT

```
      SUBROUTINE XXISRT( N , ARR , INDX , WRK )
      IMPLICIT NONE
C-----------------------------------------------------------------------
```

```
C
C  ****************** FORTRAN77 SUBROUTINE: XXISRT *********************
C
C  PURPOSE: SORTS AN ARRAY 'ARR' OF LENGTH 'N' ACCORDING TO THE INDEX
C           'INDX()'  SUCH  THAT  'ARR(INDX(J))'  GOES  TO   'ARR(J)'
C           FOR J=1,2,...,N.   THE INPUT QUANTITIES 'N' AND 'INDX' ARE
C           NOT CHANGED.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT :    (I*4)  N       = INPUT ARRAY 'ARR()' LENGTH
C  I/O   :    (R*8)  ARR()   = ARRAY TO BE SORTED
C  INPUT :    (I*4)  INDX()  = RE-ORDERING INDEX
C  OUTPUT:    (R*8)  WRK()   = WORKSPACE
C
C             (I*4)  I       = GENERAL USE
C
C ROUTINES:  NONE
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    11/07/90
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER N        , I
      INTEGER INDX(N)
C-----------------------------------------------------------------------
      REAL*8  ARR(N)   , WRK(N)
C-----------------------------------------------------------------------
```

XXLIM4

```
       SUBROUTINE XXLIM4 ( GMIN , GMAX , X , N , CUTMIN )
       IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXLIM4 *********************
C
C  PURPOSE:   FINDS MAXIMA AND MINIMA OF A LIST  OF X VALUES AND RETURNS
C             THE SCALE RANGE FOR PLOTTING ON A LOG TO THE BASE 10 GRID.
C             I.E. LOG10 RANGE TO NEAREST APPROPRIATE INTEGERS.
C             (RANGE IN LINEAR UNITS = GMIN -> GMAX)
C             (REAL VERSION OF 'XXLIM8')
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  OUTPUT: (R*4)  GMIN    = LOWER LIMIT FOR GRAPH X-AXIS (LINEAR)
C  OUTPUT: (R*4)  GMAX    = UPPER LIMIT FOR GRAPH X-AXIS (LINEAR)
C  INPUT : (R*4)  X()     = INPUT X-VALUES
C  INPUT : (I*4)  N       = NUMBER OF INPUT X-VALUES
C  INPUT : (R*4)  CUTMIN  = MINIMUM ALLOWED VALUE FOR 'GMIN' (LINEAR).
C                           ('CUTMIN=0' EFFECTIVELY REMOVES ITS EFFECT)
C                                                                  0000000
C          (I*4)  NCMAX   =  PARAMETER = MAXIMUM NUMBER OF LOG10 CYCLES
C                                        ALLOWED IN GHOST80
C          (I*4)  IMIN    = MINIMUM LOG10 CYCLE
C          (I*4)  IMAX    = MAXIMUM LOG10 CYCLE
C          (I*4)  I       = GENERAL USE
C                                                                  0000000
C AUTHOR:    PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C            K1/0/81
C            JET EXT. 4569
C
C DATE:      02/08/90
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE (NO CHANGES)
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
       INTEGER NCMAX
C-----------------------------------------------------------------------
       PARAMETER( NCMAX = 20 )
C-----------------------------------------------------------------------
       INTEGER N      , I
       INTEGER IMIN   , IMAX
C-----------------------------------------------------------------------
       REAL*4  GMIN   , GMAX   , CUTMIN
C-----------------------------------------------------------------------
       REAL*4  X(N)
C-----------------------------------------------------------------------
```

XXLIM8

```
      SUBROUTINE XXLIM8 ( GMIN , GMAX , X , N , CUTMIN )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXLIM8 *********************
C
C  PURPOSE:   FINDS MAXIMA AND MINIMA OF A LIST  OF X VALUES AND RETURNS
C             THE SCALE RANGE FOR PLOTTING ON A LOG TO THE BASE 10 GRID.
C             I.E. LOG10 RANGE TO NEAREST APPROPRIATE INTEGERS.
C             (RANGE IN LINEAR UNITS = GMIN -> GMAX)
C             (DOUBLE PRECISION VERSION OF 'XXLIM4')
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  OUTPUT: (R*4)  GMIN    = LOWER LIMIT FOR GRAPH X-AXIS (LINEAR)
C  OUTPUT: (R*4)  GMAX    = UPPER LIMIT FOR GRAPH X-AXIS (LINEAR)
C  INPUT : (R*8)  X()     = INPUT X-VALUES
C  INPUT : (I*4)  N       = NUMBER OF INPUT X-VALUES
C  INPUT : (R*4)  CUTMIN  = MINIMUM ALLOWED VALUE FOR 'GMIN' (LINEAR).
C                           ('CUTMIN=0' EFFECTIVELY REMOVES ITS EFFECT)
C                                                                  0000000
C          (I*4)  NCMAX   =  PARAMETER = MAXIMUM NUMBER OF LOG10 CYCLES
C                                        ALLOWED IN GHOST80
C
C          (I*4)  IMIN    = MINIMUM LOG10 CYCLE
C          (I*4)  IMAX    = MAXIMUM LOG10 CYCLE
C          (I*4)  I       = GENERAL USE
C                                                                  0000000
C AUTHOR:    PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C            K1/0/81
C            JET EXT. 4569
C
C DATE:      02/08/90
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE (NO CHANGES)
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER NCMAX
C-----------------------------------------------------------------------
      PARAMETER( NCMAX = 20 )
C-----------------------------------------------------------------------
      INTEGER N      , I
      INTEGER IMIN   , IMAX
C-----------------------------------------------------------------------
      REAL*4  GMIN   , GMAX   , CUTMIN                            0000000
C-----------------------------------------------------------------------
      REAL*8  X(N)
C-----------------------------------------------------------------------
```

XXLM28

```
       SUBROUTINE XXLM28 ( GMIN   , GMAX   ,
     &                     Y      ,
     &                     NDIM1  , NDIM2  ,
     &                     LMAX1  , LMAX2  ,
     &                     CUTMIN
     &                   )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXLM28 *********************
C
C  PURPOSE: FINDS MAXIMA AND MINIMA OF A 2-DIMENSIONAL ARRAY OF Y-VALUES
C           AND  RETURNS THE SCALE RANGE FOR PLOTTING  ON A LOG  TO  THE
C           BASE 10 GRID.
C           I.E. LOG10 RANGE COVERING BOTH DIMENSIONS TO THE NEAREST
C                APPROPRIATE INTEGERS.
C           (RANGE IN LINEAR UNITS = GMIN -> GMAX)
C           (REAL*8 VERSION OF 'XXLM24')
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  OUTPUT: (R*4)  GMIN    = LOWER LIMIT FOR GRAPH Y-AXIS (LINEAR)
C  OUTPUT: (R*4)  GMAX    = UPPER LIMIT FOR GRAPH Y-AXIS (LINEAR)
C
C  INPUT : (R*8)  Y(,)    = TWO-DIMENSIONAL ARRAY OF INPUT Y-VALUES.
C
C  INPUT : (I*4)  NDIM1   = FIRST 'Y(,)' ARRAY DIMENSION BOUND
C  INPUT : (I*4)  NDIM2   = SECOND 'Y(,)' ARRAY DIMENSION BOUND
C
C  INPUT : (I*4)  LMAX1   = RANGE OF FIRST 'Y(,)' ARRAY DIMENSION
C                           TO BE ASSESSED (1 -> LMAX1) (LMAX1<=NDIM1)
C  INPUT : (I*4)  LMAX2   = RANGE OF SECOND 'Y(,)' ARRAY DIMENSION
```

```
C                           TO BE ASSESSED (1 -> LMAX2) (LMAX2<=NDIM2)
C
C  INPUT : (R*4)  CUTMIN  = MINMUM VALUE FOR 'GMIN' (LINEAR).
C                           ('CUTMIN=0' EFFECTIVELY REMOVES ITS EFFECT)
C
C          (I*4)  NCMAX   = PARAMETER = MAXIMUM ALLOWED NUMBER OF LOG10
C                                       CYCLES ALLOWED IN GHOST80
C
C          (I*4)  IMIN    = MINIMUM LOG10 CYCLE
C          (I*4)  IMAX    = MAXIMUM LOG10 CYCLE
C          (I*4)  I       = GENERAL USE
C          (I*4)  J       = GENERAL USE
C
C
C NOTES:
C        A MINIMUM AVALUE FOR 'GMIN' IS SPECIFIED IN THE ARGUMENTS.
C        (SEE 'CUTMIN').
C
C ROUTINES: NONE
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     02/08/90
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE(NO CHANGES MADE)
C-----------------------------------------------------------------------
       INTEGER NCMAX
C-----------------------------------------------------------------------
       PARAMETER( NCMAX = 20 )
C-----------------------------------------------------------------------
       INTEGER NDIM1    , NDIM2
       INTEGER LMAX1    , LMAX2
       INTEGER IMIN     , IMAX
       INTEGER I        , J
C-----------------------------------------------------------------------
       REAL*4  GMIN     , GMAX     , CUTMIN
C-----------------------------------------------------------------------
       REAL*8  Y(NDIM1,NDIM2)
C-----------------------------------------------------------------------
```

XXMADD

```
      SUBROUTINE XXMADD( NR , NC , X , A , Y , B , C )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 SUBROUTINE: XXVADD *********************
C
C  PURPOSE: ADDS TWO MATRICES WITH MULTIPLIER FOR EACH.
C
C  CALLING PROGRAM: GENERAL USE.
C
C  INPUT : (I*4)  NR      = NUMBER OF ROWS IN MATRICES 'A' AND 'B'.
C  INPUT : (I*4)  NC      = NUMBER OF COLUMNS IN MATRICES 'A' AND 'B'.
C  INPUT : (R*8)  X       = FIRST MULTIPLIER.
C  INPUT : (R*8)  A(,)    = FIRST MATRIX.
C                           1ST DIMENSION: NR
C                           2ND DIMENSION: NC
C  INPUT : (R*8)  Y       = SECOND MULTIPLIER.
C  INPUT : (R*8)  B(,)    = SECOND MATRIX.
C                           1ST DIMENSION: NR
C                           2ND DIMENSION: NC
C
C  OUTPUT: (R*8)  C       = OUTPUT MATIRX.
C                           1ST DIMENSION: NR
C                           2ND DIMENSION: NC
C
C          (I*4)  I       = LOOP INDEX.
C          (I*4)  J       = LOOP INDEX.
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    10/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C-----------------------------------------------------------------------
```

```
C
C-----------------------------------------------------------------------
      INTEGER   NR      , NC
      INTEGER   I       , J
C-----------------------------------------------------------------------
      REAL*8    X       , Y
C-----------------------------------------------------------------------
      REAL*8    A(NR,NC)  , B(NR,NC)  , C(NR,NC)
C-----------------------------------------------------------------------
```

XXMCPY

```
      SUBROUTINE XXMCPY( NR , NC , A , B )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXVADD *********************
C
C  PURPOSE: COPIES ONE MATRIX TO ANOTHER.
C
C  CALLING PROGRAM: GENERAL USE.
C
C  INPUT : (I*4)  NR      = NUMBER OF ROWS IN MATRICES 'A' AND 'B'.
C  INPUT : (I*4)  NC      = NUMBER OF COLUMNS IN MATRICES 'A' AND 'B'.
C  INPUT : (R*8)  A(,)    = INPUT MATRIX.
C                           1ST DIMENSION: NR
C                           2ND DIMENSION: NC
C
C  OUTPUT: (R*8)  B(,)    = OUTPUT MATRIX.
C                           1ST DIMENSION: NR
C                           2ND DIMENSION: NC
C
C          (I*4)  I       = LOOP INDEX.
C          (I*4)  J       = LOOP INDEX.
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    11/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER   NR      , NC
      INTEGER   I       , J
C-----------------------------------------------------------------------
      REAL*8    A(NR,NC)  , B(NR,NC)
C-----------------------------------------------------------------------
```

XXMINO

```
      SUBROUTINE XXMINO( ND , A , B )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXMINO *********************
C
C  PURPOSE: INVERTS A MATRIX.
C
C  CALLING PROGRAM: GENERAL USE.
C
C  INPUT : (I*4)  ND      = DIMENSION OF MATRICES 'A' AND 'B'.
C  INPUT : (R*8)  A(,)    = MATRIX TO BE INVERTED.
C                           1ST DIMENSION: ND
C                           2ND DIMENSION: ND
C
C  OUTPUT: (R*8)  B(,)    = INVERTED MATRIX.
C                           1ST DIMENSION: ND
C                           2ND DIMENSION: ND
C
C
C          (I*4)  I       = LOOP INDEX.
C          (I*4)  J       = LOOP INDEX.
C
C          (R*8)  DINT    = DUMMY ARGUMENT TO 'XXMINV'.
C
C          (L*4)  LSOLVE  = .FALSE. => 'XXMINV' ONLY INVERTS MATRIX.
C
C  OUTPUT: (R*8)  DUM()   = DUMMY ARGUMENT TO 'XXMINV'.
```

```
C                          DIMENSION: ND
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          XXMINV     ADAS      BOTH INVERTS A MATRIX AND SOLVES A SET
C                               OF SIMULTANEOUS LINEAR EQUATIONS.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    10/11/93
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER   ND
      INTEGER   I       , J
C-----------------------------------------------------------------------
      REAL*8    DINT
C-----------------------------------------------------------------------
      LOGICAL   LSOLVE
C-----------------------------------------------------------------------
      REAL*8    A(ND,ND)  , B(ND,ND)
      REAL*8    DUM(100)
C-----------------------------------------------------------------------
```

XXMINV

```
      SUBROUTINE XXMINV( LSOLVE , NDMAX , NDIM   ,
     &                   A      , B     , DINT
     &                 )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXMINV *********************
C
C  PURPOSE: MATRIX INVERSION WITH ACCOMPANYING SOLUTION OF LINEAR
C           EQUATIONS IF REQUESTED.
C
C
C  CALLING PROGRAM: GENERAL USE
C
C
C  SUBROUTINE:
C
C  INPUT : (L*4)  LSOLVE   = .TRUE.  => SOLVES  SET  OF   'N'   LINEAR
C                                       EQUATIONS A.X = B, WHERE A,X,B
C                                       ARE MATRICES/VECTORS AND:
C                                       A = 'A(,)' ON INPUT
C                                       X = 'B()'  ON OUTPUT
C                                       B = 'B()'  ON INPUT
C                            .FALSE. => DOES NOT SOLVE ANY EQUATIONS
C                                       AND 'B()' IS NOT USED.
C
C  INPUT : (I*4)  NDMAX    = MAXIMUM VALUE OF 'N' ALLOWED. I.E.
C                             PHYSICAL DIMENSION OF INPUT MATRICES.
C  INPUT : (I*4)  NDIM     = MAXIMUM VALUE OF 'N' ALLOWED. I.E.
C                             PHYSICAL DIMENSION OF INPUT MATRICES.
C
C  I/O   : (R*8)  A(,)     = 'N' BY 'N' MATRIX
C                             INPUT : MATRIX TO BE INVERTED
C                             OUTPUT: INVERTED MATRIX
C  I/O   : (R*8)  B()      = USED ONLY IF 'LSOLVE=.TRUE.'WHEN THE SET
C                             OF 'N' LINEAR EQUATIONS ARE TO BE SOLVED.
C                             A.X = B
C                             INPUT : RIGHT HAND SIDE VECTOR 'B'
C                             OUTPUT: SOLUTION VECTOR 'X'
C
C  OUTPUT: (R*8)  DINT     = +1 OR -1 DEPENDING ON WHETHER THE  NUMBER
C                             OF  ROW  INTERCHANGES  WAS  EVEN  OR  ODD,
C                             RESPECTIVELY.
C
C          (I*4)  NLMAX    = PARAMETER: MUST BE >= 'NDMAX'
C
C          (I*4)  LROW     = ROW OF INPUT 'A(*,)' CONTAINING LARGEST
C                             NON-ZERO ELEMENT.
C          (I*4)  LCOLUM   = COLUMN OF INPUT 'A(,*)' CONTAINING LARGEST
C                             NON-ZERO ELEMENT.
C                             'LROW' & 'LCOLUM' => PIVOT POSITION IN A(,)
C          (I*4)  I1       = ARRAY INDICES: 1ST LEVEL
C          (I*4)  I2       = ARRAY INDICES: 2ND LEVEL
C          (I*4)  I3       = ARRAY INDICES: 3RD LEVEL
C
```

```
C          (R*8)  DMAX     = LARGEST NON-ZERO ELEMENT OF 'A(,)',
C                             AND EQUALS 'ABS(A(LROW,LCOLUM))'
C          (R*8)  SWAP     = TEMPORARAY STORAGE WHEN INTERCHANGING
C                             ROWS/COLUMNS.
C          (R*8)  TVAL     = TEMPORARY STORAGE OF VALUES IN NON-PIVOT
C                             ROWS.
C
C          (I*4)  IPIVOT() = NUMBER OF TIMES PIVOT FOUND IN COLUMN
C                             DIMENSION: COLUMN INDEX
C          (I*4)  INDX(,1) = ROW OF PIVOT IN 'A(,)'
C                             DIMENSION: PIVOT INDEX
C          (I*4)  INDX(,2) = COLUMN OF PIVOT IN 'A(,)'
C                             DIMENSION: PIVOT INDEX
C
C          (R*8)  PIVOT()  = 'DMAX' FOR PIVOT: DIMENSION PIVOT INDEX
C
C
C ROUTINES: NONE
C
C NOTE: THIS IS A STRUCTURED VERSION OF THE SUBROUTINE 'MATINV' WRITTEN
C       BY HUGH SUMMERS.
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    31/08/90
C
C UPDATE:  15/05/92 - PE BRIDEN - INCREASED NLMAX FROM 50 -> 75
C
C UPDATE:  26/06/92 - PE BRIDEN - INCREASED NLMAX FROM 75 -> 200
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER    NLMAX
C-----------------------------------------------------------------------
      PARAMETER( NLMAX = 200 )
C-----------------------------------------------------------------------
      INTEGER    NDMAX          , NDIM
      INTEGER    LROW           , LCOLUM        ,
     &           I1             , I2            , I3
C-----------------------------------------------------------------------
      REAL*8     DINT
      REAL*8     DMAX           , SWAP          , TVAL
C-----------------------------------------------------------------------
      LOGICAL    LSOLVE
C-----------------------------------------------------------------------
      INTEGER    IPIVOT(NLMAX)  , INDX(NLMAX,2)
C-----------------------------------------------------------------------
      REAL*8     A(NDMAX,NDMAX) , B(NDMAX)
      REAL*8     PIVOT(NLMAX)
C-----------------------------------------------------------------------
```

XXMMUL

```
      SUBROUTINE XXMMUL( NR , NRC , NC , A , B , C )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXMMUL *********************
C
C  PURPOSE: MULTIPLIES TWO MATRICES.
C
C  CALLING PROGRAM: GENERAL USE.
C
C  INPUT : (I*4)  NR      = NO. OF ROWS IN MATRICES 'A' AND 'C'.
C  INPUT : (I*4)  NRC     = NO. OF COLUMNS IN MATIX 'A' AND ROWS IN
C                           MATIRX 'B'.
C  INPUT : (I*4)  NC      = NO. OF COLUMNS IN MATRICES 'B' AND 'C'.
C  INPUT : (R*8)  A(,)    = FIRST MATRIX STORED AS A LINEAR VECTOR BY
C                           COLUMN.
C                           1ST DIMENSION: NR
C                           2ND DIMENSION: NRC
C  INPUT : (R*8)  B(,)    = SECOND MATRIX STORED AS A LINEAR VECTOR BY
C                           COLUMN.
C                           1ST DIMENSION: NRC
C                           2ND DIMENSION: NC
C
C  OUTPUT: (R*8)  C(,)    = RESULT MATRIX STORED AS LINEAR VECTOR BY
C                           COLUMN.
C                           1ST DIMENSION: NR
C                           2ND DIMENSION: NC
C                           DIMENSION: MXTERM
C
C          (I*4)  I       = LOOP INDEX.
C          (I*4)  J       = LOOP INDEX.
C          (I*4)  K       = LOOP INDEX.
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    10/11/93
```

```
C
C UNIX-IDL PORT:
C
C AUTHOR:  WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC)
C
C DATE:    22ND MAY 1996
C
C VERSION: 1.1                          DATE: 22-05-96
C MODIFIED: WILLIAM OSBORN
C               - FIRST VERSION. IBM VERSION NOT CHANGED
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER   NR      , NRC    , NC
      INTEGER   I       , J      , K
C-----------------------------------------------------------------------
      REAL*8    SUM
C-----------------------------------------------------------------------
      REAL*8    A(NR,NRC)  , B(NRC,NC)  , C(NR,NC)
C-----------------------------------------------------------------------
```

XXMNMX

```
      SUBROUTINE XXMNMX( LOGFIT , MAXDEG , TOLVAL ,
     &                   NIN    , XIN    , YIN    ,
     &                   NOUT   , XOUT   , YOUT   ,
     &                   NCOEF  , COEF ,
     &                   MINFO
     &                 )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXMNMX *********************
C
C  PURPOSE:           TO EVALUATE THE TAYLOR COEFFICIENTS OF THE MINIMAX
C                     POLYNOMIAL USING A CALL TO A NAG LIBRARY ROUTINE.
C
C                     OUTPUTS 'NOUT' INTERPOLATED COORDINATES  USING THE
C                     MINIMAX EVALUATION, WHICH ARE EVENLY  SPACED ALONG
C                     THE X-AXIS COVERING THE X-VALUE RANGE  INPUT.   IF
C                     REQUIRED THE SPACING WILL BE EVENLY  SPACED  ALONG
C                     THE LOG10 TRANSFORMATION OF THE  X-AXIS  (IN  THIS
C                     CASE THE Y-VALUES ARE SIMILARLY TRANSFORMED).
C
C                     * IMPORTANT: 'NOUT' >= 'NIN' (NO. OF INPUT VALUES)
C
C                     MINFO(CHARACTER*80) ON OUTPUT CONTAINS INFORMATION
C                     REGARDING THE SLOPE OF THE CURVE AT THE BOUNDARIES
C                     OF THE EXPERIMENTAL REGION, AND AN ESTIMATE OF THE
C                     MAXIMUM ERROR IN THE MINIMAX INTERPOLATION.
C
C  CALLING PROGRAMS:  GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (L*4)  LOGFIT  = .TRUE.  => PERFORM MINIMAX POLYNOMIAL EVAL-
C                                      UATION ON LOG10 TRANSFORMATIONS
C                                      OF THE INPUT X AND Y DATA.
C                           .FALSE  => PERFORM MINIMAX POLYNOMIAL EVAL-
C                                      UATION ON THE X AND Y VALUES AS
C                                      INPUT.
C  INPUT : (I*4)  MAXDEG  = MAXIMUM   POSSIBLE  DEGREE  OF   POLYNOMIAL
C                           ALLOWED IN THE MINIMAX FITTING (MUST BE< 26)
C                           (IF 'NOUT'<100 => MAX. DEGREES ALLOWED = 22)
C                           NOTE:
C                           'COEF()' MUST HAVE .GE. 'MAXDEG+1' ELEMENTS
C  INPUT : (R*8)  TOLVAL  = PARAMETER = FRACTIONAL TOLERANCE FOR ACCEPT-
C                                       ANCE OF DATA FITTED  BY  MINIMAX
C                                       POLYNOMIAL.  (IF IT EQUALS  ZERO
C                                       THEN RUNS TO MAX. DEGREE).
C
C  INPUT : (I*4)  NIN     = NUMBER OF INPUT KNOTS
C  INPUT : (R*8)  XIN()   = INPUT X-VALUES OF KNOTS
C  INPUT : (R*8)  YIN()   = INPUT Y-VALUES OF KNOTS
C
C  INPUT : (I*4)  NOUT    = NUMBER  OF  OUTPUT  VALUES  REQUIRED  TO  BE
C                           INTERPOLATED USING MINIMAX POLYNOMIAL EVAL-
C                           UATION - SPACED EVENLY BETWEEN THE MINIMUM
C                           AND MAXIMUM VALUES OF THE (TRANSFORMED) X-
C                           VALUES INPUT.
C  OUTPUT: (R*8)  XOUT()  = INITIALLY: ORDERED/(TRANSFORMED) 'XIN()'
C                                      (ASCENDING ORDER).
C                                      'NIN' VALUES
C                           OUTPUT   : X-VALUES FOR WHICH INTERPOLATION
C                                      IS CARRIED OUT.
C                                      'NOUT' VALUES
C  OUTPUT: (R*8)  YOUT()  = INITIALLY: ORDERED/(TRANSFORMED) 'YIN()'
C                                      (ORDERED ACCORDING TO 'XOUT()').
C                                      'NIN' VALUES
C                           OUTPUT   : INTERPOLATED Y-VALUES FOR THE
C                                      X-VALUES ('XOUT()' - OUTPUT)
C                                      'NOUT' VALUES
C
C  OUTPUT: (I*4)  NCOEF   = NUMBER OF MINIMAX FIT COEFFICIENTS
```

```
C  OUTPUT: (R*8)  COEF()  = MINIMAX COEFFICIENTS - ARRAY SIZE: MAXDEG+1
C
C  OUTPUT: (C*80) MINFO   = DIAGNOSTIC INFORMATION STRING
C
C          (I*4)  MAXOUT  = PARAMETER = MAXIMUM NUMBER OF OUTPUT X,Y CO-
C                                       ORDINATES THAT CAN BE INTERPOLA-
C                                       TED.
C          (I*4)  MAXTOL  = PARAMETER = POWER OF 10 WHICH REPRESENTS THE
C                                       MAXIMUM FRACTIONAL DIFFERENCE
C                                       ALLOWED BETWEEN ACTUAL (YDATA)
C                                       AND FITTED (YFIT) DATA.
C          (I*4)  MINTOL  = PARAMETER = POWER OF 10 WHICH REPRESENTS THE
C                                       MINIMUM FRACTIONAL DIFFERENCE
C                                       ALLOWED BETWEEN ACTUAL (YDATA)
C                                       AND FITTED (YFIT) DATA.
C
C          (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C          (I*4)  ILIMIT  = UPPER LIMIT  OF  THE  NUMBER  OF  DEGREES
C                           ALLOWED FOR MINIMAX POLYNOMIAL EVALUATION.
C          (I*4)  IDEG    = NUMBER OF DEGREES BEING USED FOR MINIMAX
C                           POLYNOMIAL EVALUATION.
C          (I*4)  NDEG    = NUMBER OF DEGREES FOR ACCEPTED MINIMAX FIT.
C          (I*4)  I1      = GENERAL ARRAY INDEX
C          (I*4)  I2      = GENERAL ARRAY INDEX
C
C          (R*8)  XMIN    = MINIMUM (TRANSFORMED) X-VALUE INPUT
C          (R*8)  XMAX    = MAXIMUM (TRANSFORMED) X-VALUE INPUT
C          (R*8)  REF     = OUTPUT FROM 'XXCHEB' = NEGATIVE IF MINIMAX
C                           FITTING PROCEDURE IS CYCLING.
C          (R*8)  SUM     = USED TO SUM UP POLYNOMIAL TERMS WHEN CALCUL-
C                           ATING VALUE OF 'Y' AT GIVEN 'X'.
C          (R*8)  YFIT    = FITTED Y-VALUE USING MINIMAX POLYNOMIAL
C          (R*8)  YDATA   = INPUT Y-VALUE FOR COMPARISON TO FITTED VALUE
C          (R*8)  DIFF    = FRACTIONAL DIFFERENCE BETWEEN ACTUAL (YDATA)
C                           AND  FITTED  (YFIT)  Y-VALUE   FROM  MINIMAX
C                           POLYNOMIAL FITTED.
C          (R*8)  BIG     = LARGEST FRACTIONAL ERROR BETWEEN ACTUAL AND
C                           FITTED Y-VALUES FOR POLYNOMIAL EVALUATION.
C          (R*8)  GRADL   = GRADIENT OF FITTED MINIMAX POLYNOMIAL CURVE
C                           AT LOWER BOUNDARY OF INPUT DATA.
C          (R*8)  GRADU   = GRADIENT OF FITTED MINIMAX POLYNOMIAL CURVE
C                           AT UPPER BOUNDARY OF INPUT DATA.
C          (R*8)  XSTEP   = SEPERATION   OF   OUTPUT  X-VALUES  TO   BE
C                           INTERPOLATED.
C          (R*8)  XVAL    = X-VALUES AT WHICH INTERPOLATION IS TAKING
C                           PLACE.
C
C          (C*6)  CFTYPE  = 'LOGFIT' IF 'LOGFIT=.TRUE.'
C                           'LINFIT' IF 'LOGFIT=.FALSE.'
C
C          (L*4)  LMFIT   = .TRUE.  => MINIMAX POLYNIOMIAL FOUND WHICH
C                                      IS WITHIN DESIRED TOLERANCE.
C                                      (SEE 'TOLVAL')
C                           .FALSE. => NO MINIMAX POLYNIOMIAL FOUND WITH
C                                      DESIRED TOLERANCE.
C                                      (SEE 'TOLVAL')
C
C          (I*4)  INDX()  = ASCENDING ORDER INDEX FOR INPUT X-VALUES
C
C          (R*8)  WRK()   = WORKING SPACE FOR ORDERING DATA
C
C
C NOTES:
C          THIS SUBROUTINE IS AN AMMENDED AND STRUCTURED VERSION OF  THE
C          SUBROUTINE  'MINIMAX'  WRITTEN BY  STEPHEN TURNER,  JET  30TH
C          JULY 1989. IT NO LONGER ORDERS THE INPUT COORDINATES DIRECTLY
C          BUT COPIES THE TO A SECOND ARRAY AND ORDERS THEM.   THEREFORE
C          IF THE INPUT COORDINATES ARE REQUIRED TO BE ORDERED THIS MUST
C          BE CARRIED OUT EXPLICITLY IN THE CALLING OPROGRAM/ROUTINE.
C          THE SUBROUTINE NO LONGERS REQUIRES THE INPUT CO-ORDINATES  IN
C          LOG10 FORM, THIS TRANSFORMATION CAN BE CARRIED OUT EXPLICITLY
C          BY THE SUBROUTINE.   THE  ROUTINE HAS BEEN WRITTEN TO  BE  OF
C          GENERAL RATHER THAN SPECIFIC USE.
C
C          IN THE ORIGINAL 'MINIMAX' SUBROUTINE THE MINIMUM NUMBER OF
C          DEGREES ACCEPTED FOR A FIT WAS 2, IN 'XXMNMX' IT IS 1.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          XXINDX     ADAS     GIVES INDICES OF SORTED ARRAY (ASCENDNG)
C          XXISRT     ADAS     RE-ORDERS ARRAY ACCORDING TO 'XXINDX'
C          XXCHEB     ADAS     MINIMAX POLYNOMIAL COEFT. EVALUATION
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     12/10/90
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C UPDATE:  14/02/94 - PE BRIDEN - ADAS91: CORRECTED BUG WHEN CHECKING
C                                         TOLERANCE FOR LOGFIT - CHANGED
```

```
C                                       THE LINE:
C                                               YDATA = YIN(I1)
C                                       TO:
C                                               YDATA = YIN(INDX(I1))
C                                       (WOULD CAUSE PROBLEMS IF YIN
C                                        NOT ORDERED ON INPUT.)
C
C UPDATE:  14/02/94 - PE BRIDEN - ADAS91: RECODED SECTION USE TO TEST
C                                       TOLERANCE VALUE - NOW CHECKS
C                                       FOR DIVISION BY 0 AND USES
C                                       LOG10 TO STORES VALUES THUS
C                                       AVOIDING OVERFLOWS.
C                                       (BIG IS NOW STORED AS LOG10
C                                        VALUES INITIALLY)
C                                       INTRODUCED: MAXTOL & MINTOL
C                                               - / -
C                                       STOP EXECUTING XXCHEB IF
C                                       CYCLING (CHECK REF VARIABLE)
C                                               - / -
C                                       INITIALISED NDEG AND BIG TO
C                                       STOP ICA WARNING.
C
C UPDATE:  31/10/94 - PE BRIDEN - ADAS91: REPLACED CALL TO NAG ROUTINE
C                                       E02ACF WITH A CALL TO THE NEW
C                                       EQUIVALENT ADAS ROUTINE CALLED
C                                       XXCHEB (IDENTICAL ARGUMENTS).
C UPDATE:  17/1/95 - L JALOTA - IDL-ADAS : MODIFIED ERROR CHECKING
C                                          SECTION TO WRITE TO UNIT 0
C                                          NOT STDOUT WHICH INTERFERES
C                                          WITH PIPE COMMUNICATIONS.
C       :   6/3/95 - L JALOTA - IDL-ADAS : REPLACED CALL TO I4UNIT.
C
C UNIX-IDL PORT:
C
C VERSION: 1.2                          DATE: 10-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - INCREASED PARAMETER MAXOUT FROM 100 TO 120
C                 PRIMARILY TO PREVENT PROBLEMS WITH ADAS 506 WHERE
C                 101 POINTS ARE REQUIRED.
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER    MAXOUT         , MAXTOL      , MINTOL
C-----------------------------------------------------------------------
      PARAMETER( MAXOUT = 120   , MAXTOL = 10 , MINTOL = -10 )
C-----------------------------------------------------------------------
      INTEGER    I4UNIT
      INTEGER    MAXDEG         , NIN         , NOUT      , NCOEF
      INTEGER    ILIMIT         , IDEG        , NDEG      , I1       ,
     &           I2
C-----------------------------------------------------------------------
      REAL*8     TOLVAL
      REAL*8     XMIN           , XMAX        , BIG       , REF      ,
     &           SUM            , YFIT        , YDATA     , DIFF     ,
     &           GRADL          , GRADU       , XSTEP     , XVAL
C-----------------------------------------------------------------------
      LOGICAL    LOGFIT         , LMFIT
C-----------------------------------------------------------------------
      CHARACTER  CFTYPE*6       , MINFO*80
C-----------------------------------------------------------------------
      INTEGER    INDX(MAXOUT)
C-----------------------------------------------------------------------
      REAL*8     XIN(NIN)       , YIN(NIN)    ,
     &           XOUT(NOUT)     , YOUT(NOUT)  ,
     &           COEF(MAXDEG+1)
      REAL*8     WRK(MAXOUT)
C
C-----------------------------------------------------------------------
```

XXOPEN

```
       SUBROUTINE XXOPEN( IUNIT, DSFULL , LEXIST )
       IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXOPEN *********************
C
C  PURPOSE: TO INQUIRE & OPEN INPUT DATA FILE & ALLOCATE TO UNIT 'IUNIT'
C           (READ ONLY)  - IF IT DOES NOT EXISTS A MESSAGE IS  SENT  TO
C           THE SCREEN AND LEXIST IS RETURNED AS FALSE.
C
C  CALLING PROGRAM:  GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT :    (I*4)  IUNIT   = UNIT TO WHICH INPUT FILE IS ALLOCATED
C  INPUT :    (C*(*))DSFULL  = FULL INPUT DATA SET NAME (INCL. USERID)
C                              IN FORM FOR DYNAMIC ALLOCATION.
C  OUTPUT:    (L*4)  LEXIST  = .TRUE.  => DATA SETS EXISTS AND IS OPEN
C                              = .FALSE. => DATA SET DOES NOT EXIST
C
C             (I*4)  I4UNIT  = FUNCTION (SEE ROUTINE SECTION BELOW)
C             (C*1)  BSLASH  = '/' - MUST BE FIRST 'DSFULL'CHARACTER
C NOTE:
```

```
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4UNIT     ADAS      FETCH UNIT NUMBER FOR OUTPUT OF MESSAGES
C          XXTERM     ADAS      TERMINATES ADAS PROGRAM WITH MESSAGE
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 2520
C
C DATE:    04/03/91 - ADAS91 VERSION (INCLUDES INQUIRE)
C
C UPDATE:  07/08/91 - PE BRIDEN: ADDED ERROR HANDLING FOR OPEN ERROR.
C
C UPDATE:  23/04/93 - PE BRIDEN - ADAS91: ADDED I4UNIT FUNCTION TO WRITE
C                                         STATEMENTS FOR SCREEN MESSAGES
C
C UPDATE:  24/05/93 - PE BRIDEN - ADAS91: CHANGED I4UNIT(0)-> I4UNIT(-1)
C
C-----------------------------------------------------------------------
       INTEGER   I4UNIT
       INTEGER   IUNIT
C-----------------------------------------------------------------------
       LOGICAL   LEXIST
C-----------------------------------------------------------------------
       CHARACTER DSFULL*(*)   , BSLASH*1
C-----------------------------------------------------------------------
```

XXRATE

```
      SUBROUTINE XXRATE( NDTRN  , NDTEM  , NDLEV ,
     &                   ICNT   , MAXT   ,
     &                   XJA    , ER     , TEMP  ,
     &                   I1A    , I2A    ,
     &                   RATE1  , DRATE1
     &                 )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXRATE *********************
C
C  PURPOSE: TO CALCULATE THE EXCITATION AND DE-EXCITATION RATE COEFFICI-
C           ENTS FOR A SET OF INPUT TEMPERATURES(rydberg) & TRANSITIONS.
C           VALUES RETURNED ASSUMING UNIT GAMMA VALUES (I.E. GAMMA = 1)
C
C  CALLING PROGRAM:  GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT :  (I*4)  NDTRN   = MAXIMUM NUMBER OF TRANSITIONS ALLOWED
C  INPUT :  (I*4)  NDTEM   = MAXIMUM NUMBER OF TEMPERATURES ALLOWED
C  INPUT :  (I*4)  NDLEV   = MAXIMUM NUMBER OF ENERGY LEVELS ALLOWED
C
C  INPUT :  (I*4)  ICNT    = NUMBER OF TRANSITIONS
C  INPUT :  (I*4)  MAXT    = NUMBER OF ISPF INPUT TEMPERATURES
C
C  INPUT :  (R*8)  XJA()   = QUANTUM NUMBER (J-VALUE) FOR GIVEN LEVEL.
C                            NOTE: (2*XJA)+1 = STATISTICAL WEIGHT
C  INPUT :  (R*8)  ER()    = ENERGY RELATIVE TO LEVEL 1 (RYDBERGS)
C                            DIMENSION ENERGY LEVEL.
C  INPUT :  (R*8)  TEMP()  = ISPF READ TEMPERATURES (KELVIN)
C
C
C  INPUT :  (I*4)  I1A()   = TRANSITION: LOWER ENERGY LEVEL INDEX
C  INPUT :  (I*4)  I2A()   = TRANSITION: UPPER ENERGY LEVEL INDEX
C
C  OUTPUT:  (R*8)  RATE1(,) = UNIT GAMMA EXCITATION RATE COEFFS(cm**3/s)
C                             1st DIMENSION: TEMPERATURE INDEX
C                             2nd DIMENSION: TRANSITION INDEX
C  OUTPUT:  (R*8)  DRATE1(,)= UNIT GAMMA DE-EXCIT'N RATE COEFFS(cm**3/s)
C                             1st DIMENSION: TEMPERATURE INDEX
C                             2nd DIMENSION: TRANSITION INDEX
C
C           (I*4)  NLTEM   = PARAMETER = MUST BE >= 'NDTEM'
C
C           (R*8)  TK2ATE  = PARAMETER = EQUATION CONSTANT = 1.5789D+05
C           (R*8)  R2GAM   = PARAMETER = EQUATION CONSTANT = 2.17161D-08
C
C           (I*4)  IC      = TRANSITION ARRAY INDEX
C           (I*4)  IT      = TEMPERATURE ARRAY INDEX
C
C           (R*8)  EUPPER  = SELECTED TRANSITION - UPPER ENERGY
C                            LEVEL RELATIVE TO INDEX LEVEL 1 (Rydbergs)
C           (R*8)  ELOWER  = SELECTED TRANSITION - LOWER ENERGY
C                            LEVEL RELATIVE TO INDEX LEVEL 1 (Rydbergs)
C           (R*8)  WUPPER  = SELECTED TRANSITION - UPPER ENERGY LEVEL
C                            STATISTICAL WEIGHT.
C           (R*8)  WLOWER  = SELECTED TRANSITION - LOWER ENERGY LEVEL
C                            STATISTICAL WEIGHT.
C           (R*8)  SUPPER  = 1/(UPPER LEVEL STATISTICAL WEIGHT)
C           (R*8)  SLOWER  = 1/(LOWER LEVEL STATISTICAL WEIGHT)
C           (R*8)  RYDDIF  = NEGATIVE TRANSITION ENERGY IN RYDBERGS
C                            ( NOTE: 1 Rydberg = 1.09737E5 cm-1)
C
```

```
C           (R*8)  ATE()   = EQUATION PARAMETER: DIMENSION - TEMPERATURE
C           (R*8)  GVAL()  = EQUATION PARAMETER (FOR UNIT GAMMA VALUE)
C                            DIMENSION - TEMPERATURE
C
C ROUTINES:  NONE
C
C NOTES:
C            EQUATIONS USED -
C
C                          2.17161E-8 x GAMMA x SQRT(157890 / TEMP)
C            RATE1 = ------------------------------------------------
C                    WLOWER x EXP( (EUPPER-ELOWER) * (157890 / TEMP) )
C
C                          2.17161E-8 x GAMMA x SQRT(157890 / TEMP)
C            DRATE1 =  ----------------------------------------
C                                       WUPPER
C
C            NOTE: OUTPUT VALUES ARE FOR 'GAMMA=1'
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    18/09/90
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER    NLTEM
C-----------------------------------------------------------------------
      REAL*8     TK2ATE             , R2GAM
C-----------------------------------------------------------------------
      PARAMETER( NLTEM = 101        )
C-----------------------------------------------------------------------
      PARAMETER( TK2ATE = 1.5789D+5 , R2GAM = 2.17161D-8 )
C-----------------------------------------------------------------------
      INTEGER    NDTRN              , NDTEM              , NDLEV       ,
     &           ICNT               , MAXT
      INTEGER    IC                 , IT
C-----------------------------------------------------------------------
      REAL*8     EUPPER             , ELOWER             ,
     &           WUPPER             , WLOWER             ,
     &           SUPPER             , SLOWER             , RYDDIF
C-----------------------------------------------------------------------
      INTEGER    I1A(NDTRN)         , I2A(NDTRN)
C-----------------------------------------------------------------------
      REAL*8     XJA(NDLEV)         , ER(NDLEV)          , TEMP(NDTEM) ,
     &           RATE1(NDTEM,NDTRN) , DRATE1(NDTEM,NDTRN)
      REAL*8     ATE(NLTEM)         , GVAL(NLTEM)
C-----------------------------------------------------------------------
```

XXREIA

```
      SUBROUTINE XXREIA( IUNIT , LSWIT , EIA )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXREIA *********************
C
C  PURPOSE: TO READ IN IONISATION POTENTIALS FROM AN INPUT MASTER
C           CONDENSED FILE ALLOCATED TO UNIT 'IUNIT' WITHOUT KNOWING HOW
C           MANY ARE PRESENT.
C
C  CALLING PROGRAMS: XXIN17 , XXIN80 , XXINST & GENERAL USE.
C
C  SUBROUTINE:
C
C  INPUT : (I*4)  IUNIT   = UNIT TO WHICH CONDENSED FILE IS ALLOCATED
C  OUTPUT: (L*4)  LSWIT   = .TRUE.  => VALID SET OF 'EIA()' VALUES FOUND
C                           .FALSE. =>INVALID OR NO 'EIA()' VALUES FOUND
C  OUTPUT: (R*8)  EIA()   = IONISATION RATE COEFFICIENTS: ()=ION CHARGE
C                           UNITS - WAVE NUMBERS (CM-1)
C
C          (I*4)  IPOT    = INDEX OF IONISATION RATE COEFFT. VALUE BEING
C                           ANALYSED.
C          (I*4)  IREC    = NUMBER RECORDS READ IN FROM 'IUNIT'
C          (I*4)  IBGN    = FIRST BYTE IN 'STRING' CONTAINING THE 'EIA()
C                           VALUE BEING ANALYSED.
C          (I*4)  IEND    = LAST BYTE IN 'STRING' CONTAINING THE 'EIA()'
C                           VALUE BEING ANALYSED.
C                           (NOTE: 'EIA()' VALUES IF PRESENT ARE STORED
C                                  SIX TO A LINE AS 6F12.1)
C          (I*4)  IABT    = RETURN CODE FROM 'R8FCTN' FUNCTION.
C                           0 => NO ERROR
C                           2 => INVALID FLOATING POINT NUMBER
C                           (A BLANK INPUT STRING TO 'R8FCTN' CAUSES
C                            A ZERO VALUE TO BE RETURNED AND IABT=0)
C          (I*4)  I       = GENERAL ARRAY USE
C
C          (L*4)  LDEND   = .TRUE. =>VALID END TO 'EIA()' SECTION FOUND
C                           .FALSE.=>INVALID END TO 'EIA()'SECTION FOUND
C
C          (C*12) C12     = 12 BYTE STRING FOR STORING 'EIA()' VALUES
C          (C*80) STRING  = 80 BYTE STRING FOR STORING INPUT FILE RECORD
C
```

```
C
C NOTE:
C           ON RETURN THE NEXT RECORD SHOULD BE THAT CONTAINING THE
C           REDUCED DENSITY VALUES.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          R8FCTN     ADAS      FUNCTION - CONVERT STRING -> REAL*8 NUM
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    22/08/90
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 06-09-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE (NO CHANGES)
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER   IUNIT   , IPOT      , IREC    , IBGN     , IEND     ,
     &          I       , IABT
C-----------------------------------------------------------------------
      REAL*8    R8FCTN
C-----------------------------------------------------------------------
      LOGICAL   LSWIT   , LDEND
C-----------------------------------------------------------------------
      CHARACTER C12*12  , STRING*80
C-----------------------------------------------------------------------
      REAL*8    EIA(50)
C-----------------------------------------------------------------------
```

XXSIM

```
      SUBROUTINE XXSIM(A,M,B,N,X,WKS,ERR,ISTOP)
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C ROUTINE: XXSIM
C
C PURPOSE: SOLVES THE SYSTEM OF SIMULTANEOUS EQUATIONS AX=B USING THE
C           NETLIB LINALG ROUTINE LSQR WHICH FOLLOWS THIS ROUTINE ALONG
C          WITH ITS DEPENDENCIES. THIS ROUTINE REPLACES NAG LIBRARY
C          ROUTINE F04ATF. HOWEVER, THE LU DECOMPOSITION IS NOT OUTPUT.
C
C CALLING PROGRAM: GENERAL USE
C
C INPUT:
C
C     (EXT.)  MA      SUBROUTINE TO CALCULATE PRODUCTS OF A WITH GIVEN
C                     VECTORS. GIVEN NEXT IN THE FILE.
C     (I*4)   M       THE ROW DIMENSION OF THE TWO-DIMENSIONAL
C                     ARRAY A. NB THE MATRIX ASSUMED TO BE SQUARE AND OF
C                     ORDER N
C     (R*8)   B()     RIGHT-HAND-SIDE VECTOR B, DIMENSION = N
C     (I*4)   N       ORDER OF MATRIX A
C     (R*8)   WKS()   WORKSPACE VECTOR, DIMENSION >= N*N+2*N
C
C OUTPUT:
C
C     (R*8)   X()     SOLUTION VECTOR X, DIMENSION = N
C     (R*8)   ERR()   VECTOR OF THE ERROR ESTIMATES OF THE COMPONENTS OF
C                     X. SEE VARIABLE 'SE' IN LSQR.
C     (I*4)   ISTOP   AN ERROR CODE SET TO 0 OR 4 IF NO ERROR. SEE LSQR
C                     FOR A DESCRIPTION OF THE ERROR CODES.
C
C ROUTINES:
C-----------------------------------------------------------------------
C     NAME     SOURCE   PURPOSE
C-----------------------------------------------------------------------
C     LSQR     NETLIB   CALCULATES THE SOLUTION - SEE BELOW
C-----------------------------------------------------------------------
C
C AUTHOR: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC.)
C
C DATE:   10-06-96
C
C VERSION 1.1                             DATE: 10-06-96
C MODIFIED: WILLIAM OSBORN
C           - FIRST VERSION
C-----------------------------------------------------------------------
C
      INTEGER M, N, ISTOP
      REAL*8 A(M,N), B(N), X(N), WKS(N*N+2*N), ERR(N)
      INTEGER DUMMYI(1),ITNLIM, ITN
      REAL*8 ZERO,ONE,ANORM,ACOND,RNORM,ARNORM,XNORM,IWK
      INTEGER I,J,DIAGNOSE
      EXTERNAL MA
C-----------------------------------------------------------------------
      SUBROUTINE XXSION( SYMB , IZ , SION , LEN )
C
      IMPLICIT NONE
```

```
C
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXSION *********************
C
C  PURPOSE: RETURNS ION ELEMENT SYMBOL AND ION CHARGE AS A STRING
C           CONSTRUCTED AS FOLLOWS <SYMBOL><CHARGE>. IT ALSO RETURNS
C           THE LENGTH OF THE STRING.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (C*2)  SYMB      = ION ELEMENT SYMBOL OF BEAM.
C  INPUT : (I*4)  IZ        = ION CHARGE
C
C  INPUT : (C*4)  SION      = ION STRING '<SYMBOL><CHARGE>'.
C  INPUT : (I*4)  LEN       = LENGTH OF ION STRING.
C
C          (I*4)  I         = LENGTH IN BYTES OF NON-BLANK 'DSNAME'.
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH  (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/87
C          JET EXT. 5183
C
C DATE:    21/12/93
C
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 16-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST VERSION
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER    IZ      , LEN
C-----------------------------------------------------------------------
      CHARACTER  SYMB*2  , SION*4
C-----------------------------------------------------------------------
```

XXSLEN

```
      SUBROUTINE XXSLEN( CSTRNG , IFIRST , ILAST )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXSLEN *********************
C
C  PURPOSE: TO IDENTIFY THE FIRST AND LAST NON-BLANK CHARACTER IN A
C           STRING. (IF INPUT STRING IS BLANK IFIRST=ILAST=0)
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (C*(*)) CSTRNG   = INPUT STRING FOR INTERROGATION
C
C  OUTPUT: (I*4)   IFIRST   = BYTE POSITION OF FIRST NON-BLANK CHARACTER
C                             IN INPUT STRING.
C  OUTPUT: (I*4)   ILAST    = BYTE POSITION OF LAST  NON-BLANK CHARACTER
C                             IN INPUT STRING.
C
C          (I*4)   I        = GENERAL USE
C          (I*4)   ILEN     = LENGTH OF 'CSTRNG' STRING IN BYTES
C
C ROUTINES: NONE
C
C NOTE:
C
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 6023
C
C DATE  :  06/07/93
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER     IFIRST     , ILAST    , ILEN    , I
C-----------------------------------------------------------------------
      CHARACTER   CSTRNG*(*)
C-----------------------------------------------------------------------
      SUBROUTINE XXSPEC( USRGRP , USRTYP , USREXT )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXSPEC *********************
C
C  PURPOSE: ADAS ROUTINE - SETS UP THE DEFAULT USEGRP, USRTYP AND USREXT
C           WHICH IDENTIFY THE FILENAME AND EXTENSION TO BE READ IN
```

```
C           SUBROUTINE SPEC.  IT WORKS IN THE SAME MANNER AS XXUID WHICH
C           WHICH ALLOWS THE DEFAULT USER SPACE TO BE SET
C
C
C           USRGRP: VALUE ON INPUT  =>  USRGRP: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA GROUPNAME
C                     *                 DEFAULT ADAS DATA GROUPNAME
C                   <BLANK>             *** USRGRP VALUE NOT CHANGED ***
C                   <OTHER>             *** USRGRP VALUE NOT CHANGED ***
C
C
C           USRTYP: VALUE ON INPUT  =>  USRTYP: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA TYPENAME
C                     *                 DEFAULT ADAS DATA TYPENAME
C                   <BLANK>             *** USRTYP VALUE NOT CHANGED ***
C                   <OTHER>             *** USRTYP VALUE NOT CHANGED ***
C
C
C           USREXT: VALUE ON INPUT  =>  USREXT: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA MEMBER EXTENS.
C                     *                 DEFAULT ADAS DATA MEMBER EXTENS
C                   <BLANK>             *** USREXT VALUE NOT CHANGED ***
C                   <OTHER>             *** USREXT VALUE NOT CHANGED ***
C
C         ? => QUERIES CURRENT ADAS DATA USRGRP, USRTYP OR USREXT
C              SETTING.
C         * => SETS ADAS DATA USEGRP, USRTYP OR USREXT SETTING
C              TO DEFAULT VALUE.
C   <BLANK> => SETS ADAS DATA USRGRP, USRTYP OR USREXT SETTING
C              TO DEFAULT VALUE.
C   <OTHER> => SETS ADAS DATA USRGRP, USRTYP OR USREXT SETTING
C              TO INPUT   VALUE.
C
C  CALLING PROGRAM: SPEC AND MAIN PROGRAMS USING SPEC, ADAS503
C
C  SUBROUTINE:
C
C  I/O   : (C*8)  USRGRP   = USRFIL UNDER WHICH ADAS DATA IS STORED
C                            (IF BLANK DEFAULTS TO DEFGRP)
C
CA I/O   : (C*80) USRTYP   = SUBDIRECTORY (OPTIONAL) WHERE ADAS DATA
CA                           FILE IS LOCATED. (IF BLANK DEFAULTS TO
CA                            DEFTYP)
C
C  I/O   : (C*3)  USREXT   = USREXT UNDER WHICH ADAS DATA IS STORED
C                            (IF BLANK DEFAULTS TO DEFEXT)
C
C           (C*8)  DEFGRP   = PARAMETER = DEFAULT USER GROUP FOR ADAS

C                                         DATA SOURCE
C
C           (C*80) DEFTYP   = PARAMETER = DEFAULT SUBDIRECTORY OF ADAS
C                                         DATA SOURCE
C
C           (C*3)  DEFEXT   = PARAMETER = DEFAULT USER EXTENSION FOR ADAS
C                                         DATA SOURCE
C
C           (C*8)  ADASGR   = CURRENT ADAS DATA SOURCE GROUP
CA          (C*80) ADASTY   = CURRENT ADAS DATA SOURCE TYPE
C           (C*3)  ADASEX   = CURRENT ADAS DATA SOURCE EXTENSION
C
C
C ROUTINES:
C           ROUTINE    SOURCE    BRIEF DESCRIPTION
C           ------------------------------------------------------------
C
C NOTE:
C           TO CHECK CURRENT ADAS SOURCE USRGRP, USRTYP AND USREXT
C           CALL XXSPEC WITH ?`S AS INPUTS.
C
C AUTHOR:  HUGH P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:     2/09/93
C
C UPDATE:  L. JALOTA - 1/11/94    (TESSELLA SUPPORT SERVICES PLC)
C                      CHANGED VALUES OF DEFGRP,DEFTYP, DEFEXT SUITABLE
C                      FOR DEC ALPHA DIRECTORY STRUCTURE.
C
C UPDATE:  L.JALOTA - 23/11/94 : TIDIED UP STRING LENGTH DEFINITIONS.
C-----------------------------------------------------------------------
       CHARACTER  DEFGRP*8             , DEFTYP*80     , DEFEXT*3
C-----------------------------------------------------------------------
      PARAMETER  (DEFGRP = 'ionelec' , DEFTYP = ' ' ,
     &            DEFEXT = 'pec')
C-----------------------------------------------------------------------
      CHARACTER  USRGRP*8             , ADASGR*8      ,
     &           USRTYP*80            , ADASTY*80     ,
     &           USREXT*3             , ADASEX*3
C-----------------------------------------------------------------------
```

XXSPLE

```
      SUBROUTINE XXSPLE( LSETX , IOPT   , FINTX ,
     &                   NIN   , XIN    , YIN   ,
     &                   NOUT  , XOUT   , YOUT  ,
     &                   DY    , LINTRP
     &                 )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 SUBROUTINE: XXSPLE *********************
C
C  PURPOSE:           TO INTERPOLATE/EXTRAPOLATE USING CUBIC SPLINES
C                     (IF IOPT < 0 NO EXTRAPOLATION TAKES PLACE = VALUES
C                      SET TO ZERO).- LOGICAL ARRAY 'LINTRP()' SPECIFIES
C                      WHETHER OUTPUT SPLINE IS INTERPOLATED '.TRUE.' OR
C                      EXTRAPOLATED '.FALSE.'.
C
C                      (AS FOR 'XXSPLN' EXCEPT 'LINTRP' ARGUMENT ADDED).
C
C  CALLING PROGRAMS:  GENERAL USE
C
C  SUBROUTINE:
C
C  I/O   : (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           ( 'LSETX' IS ALWAYS RETURN AS '.FALSE.'  ON
C                             RETURN FROM THE SUBROUTINE ).
C                           ** IMPORTANT: SEE NOTES BELOW ON 'LSETX' **
C  INPUT : (I*4)  IOPT    = SPLINE END CONDITIONS/EXTRAPOLATION CONTROL
C                           SWITCH - SEE NOTES BELOW
C                           I.E. DEFINES THE BOUNDARY DERIVATIVES.
C                                (VALID VALUES = 0, 1, 2, 3, 4)
C                           IF IOPT < 0 THEN NO EXTRAPOLATION TAKES
C                           - ANY VALUES REQUIRING EXTRAPOLATION WILL BE
C                             SET TO ZERO (END CONDITIONS AS FOR IOPT=0)
C  INPUT : (R*8)  FINTX   = INTERPOLATING X-COORDINATE TRANSFORMATION.
C                           EXTERNAL FUNCTION (SEE ROUTINES BELOW)
C
C  INPUT : (I*4)  NIN     = NUMBER OF KNOTS
C  INPUT : (R*8)  XIN()   = X-VALUES OF KNOTS
C  INPUT : (R*8)  YIN()   = Y-VALUES OF KNOTS
C
C  INPUT : (I*4)  NOUT    = NUMBER OF OUTPUT VALUES TO BE INTERPOLATED
C                           EXTRAPOLATED.
C  INPUT : (R*8)  XOUT()  = X-VALUES AT WHICH INTERPOLATION/EXTRAPOLA-
C                           TION REQUIRED
C  OUTPUT: (R*8)  YOUT()  = INTERPOLATED/EXTRAPOLATED Y-VALUES FOR
C                           REQUESTED 'XOUT()' VALUES.
C
C  OUTPUT: (R*8)  DY()    = DERIVATIVES AT INPUT KNOTS (ARRAY SIZE: NIN)
C  OUTPUT: (L*4)  LINTRP()= .TRUE.  => 'YOUT()' VALUE INTERPOLATED.
C                           .FALSE. => 'YOUT()' VALUE EXTRAPOLATED.
C                           (ARRAY SIZE: NOUT)
C
C          (I*4)  NKNOTS  = PARAMETER = MAXIMUM  NUMBER OF KNOTS ALLOWED
C          (I*4)  NIOPT   = PARAMETER = MAXIMUM  VALUE OF IOPT ALLOWED
C
C          (I*4)  I       = GENERAL ARRAY USE
C          (I*4)  K       = INDEX OF 'XOUT()' VALUE FOR INTERPOLATION/
C                           EXTRAPOLATION.
C          (I*4)  NIN0    = 'NIN' - 1
C          (I*4)  INTER   = INDEX OF CLOSEST/NEXT HIGHEST VALUE OF
C                           'XIN()' TO THE VALUE OF 'XOUT()' BEING
C                           INTERPOLATED/EXTRAPOLATED.
C          (I*4)  NOPT    = VALUE OF  'IOPT' USED IN  CALCULATING  END-
C                           CONDITIONS   FOR  STORED  'X-VALUE'  SPLINE
C                           PARAMETERS.   (NOTE:  IF  'IOPT < 0',  THEN
C                           'NOPT = 0'.) - I.E. 'NOPT = MAX( 0, IOPT )'.
C
C          (R*8)  XK      = VALUE OF 'XOUT(K)' BEING INTERPOLATED/
C                           EXTRAPOLATED
C          (R*8)  XKK     = TRANSFORMED VALUE OF 'XOUT(K)' BEING
C                           INTERPOLATED/EXTRAPOLATED.
C          (R*8)  T1      = INVERSE OF SEPARATION OF KNOTS EITHER
C                           SIDE OF CURRENT KNOT.
C          (R*8)  T2      = (CURRENT KNOT POSITION TO NEXT HIGHEST KNOT
C                            POSITION) DIVIDED BY 'T1'
C          (R*8)  T3      = (CURRENT KNOT POSITION TO NEXT LOWEST  KNOT
C                            POSITION) DIVIDED BY 'T1'
C          (R*8)  T4      = INTERPOLATION FACTOR FOR CURRENT KNOT
C          (R*8)  DL1     = (REQUESTED 'XOUT()' VALUE TO NEXT HIGHEST
C                            KNOT POSITION) DIVIDED BY SEPERATION OF
C                            KNOTS EITHER SIDE OF 'XOUT(K)'.
C          (R*8)  DL2     = (REQUESTED 'XOUT()' VALUE TO NEXT LOWEST
C                            KNOT POSITION) DIVIDED BY SEPERATION OF
C                            KNOTS EITHER SIDE OF 'XOUT(K)'.
C          (R*8)  DL2     = (REQUESTED 'XOUT()' VALUE TO NEXT LOWEST
C          (R*8)  DL3     =  SEPERATION OF KNOTS EITHER SIDE OF
C                            'XOUT(K)' * 'DL1' * 'DL2'.
C
C          (L*4)  LEXTRP  = .TRUE.  => 'EXTRAPOLATION SWITCHED ON'.
```

```
C                           .FALSE. => 'EXTRAPOLATION SWITCHED OFF'.
C
C          (R*8)  QVAL()  = VALUE OF 'Q(1)'   : FUNCTION OF 'NOPT'
C          (R*8)  D2VAL() = VALUE OF 'D2(1)'  : FUNCTION OF 'NOPT'
C          (R*8)  D3VAL() = VALUE OF 'D3(1)'  : FUNCTION OF 'NOPT'
C          (R*8)  UVAL() =  VALUE OF 'U(NIN)' : FUNCTION OF 'NOPT'
C          (R*8)  AGRL() =  POLYNOMIAL CONSTANTS FOR CUBIC SPLINE FOR
C                           GIVEN 'XOUT(K)' VALUE.
C          (R*8)  X()    =  TRANSFORMED VALUES OF 'XIN()'
C          (R*8)  H()    =  SEPERATION, ALONG X-AXIS, OF KNOT FROM NEXT
C                           HIGHEST KNOT.
C          (R*8)  Q()    =  SECOND DERIVATIVE FOR KNOT
C          (R*8)  U()    =  TEMPORARY STORAGE OF DECOMPOSED FACTORS
C          (R*8)  DELY() =  SEPERATION, ALONG Y-AXIS, OF KNOT FROM NEXT
C                           HIGHEST KNOT.
C          (R*8)  D1()   =  MULTIPLICATION FACTOR USED IN CALCULATING
C                           'U()'.
C          (R*8)  D2()   =  MULTIPLICATION FACTOR USED IN CALCULATING
C                           'U()'.
C          (R*8)  D3()   =  MULTIPLICATION FACTOR USED IN CALCULATING
C                           'U()'.
C
C          (L*4)  LUVAL()=  .TRUE. => VALUE OF 'UVAL()' REFERS TO RATE
C                                     OF CHANGE OF SLOPE AT FINAL POINT.
C                           .FALSE.=> VALUE OF 'UVAL()' REFERS TO FINAL
C                                     SLOPE
C                            FUNCTION OF 'NOPT'
C
C NOTES: 'LSETX': SET TO .TRUE. ON ENTRY IF A NEW 'XIN' ARRAY IS BEING
C                 USED.  IF THE 'XIN' AXIS IS THE SAME FOR A NUMBER OF
C                 CALLS THEN DO NOT RESET 'LSETX'  -  THIS  SUBROUTINE
C                 SETS IT TO .FALSE. FOR YOU.   IF THE VALUE OF 'NOPT'
C                 IS CHANGED BETWEEN CALLS THEN THE VALUE  OF  'LSETX'
C                 ON  ENTRY IS TAKEN AS BEING EQUAL TO .TRUE. .
C
C                 THEREFORE 'LSETX' NEED ONLY BE SET TO .TRUE. ON ENTRY
C                 IF EITHER IT IS ITS FIRST CALL OR IF ANY ONE  OF  THE
C                 FOLLOWING VALUES HAS CHANGED:
C
C                 'NIN' , 'FINTX' , 'XIN(I), I=1,NIN'
C
C                 CARE: A VARIABLE MUST BE USED FOR 'LSETX', A CONSTANT,
C                       I.E.  .TRUE. ,  CANNOT BE DIRECTLY TYPED AS  AN
C                       ARGUMENT BECAUSE IT WILL BE CHANGED TO  .FALSE.
C                       ON RETURN.
C
C         SPLINE  END CONDITIONS AND EXTRAPOLATION DEPEND ON 'IOPT' AS
C         FOLLOWS:
C
C         ------------------------------------------------------------
C         | IOPT  | NOPT |  DY(1)  DDY(1)  |  DY(N)   DDY(N)  |EXTRAP'N|
C         |-------|------|-----------------|------------------|--------|
C         | < 0   |   0  |    -     0.0    |    -      0.0    |  NO    |
C         |   0   |   0  |    -     0.0    |    -      0.0    |  YES   |
C         |   1   |   1  |    -     0.0    |  -1.5      -     |  YES   |
C         |   2   |   2  |   0.0     -     |   1.0      -     |  YES   |
C         |   3   |   3  |  -0.5     -     |  -1.5      -     |  YES   |
C         |   4   |   4  |   0.0     -     |    -      0.0    |  YES   |
C         ------------------------------------------------------------
C
C            NB. OPTIONS TO BE EXTENDED FOR POWER AND CX APPLICATION
C
C         ------------------------------------------------------------
C          IF ( IOPT.LT.0 ) - NO EXTRAPOLATION TAKES PLACE VALUES SET
C                             TO ZERO (CARE IF LOG OF OUTPUT IS NEEDED).
C          IF ( IOPT.GT.4 ) PROGRAM STOPS
C         ------------------------------------------------------------
C
C          THIS SUBROUTINE IS AN AMENDED  AND STRUCTURED VERSION OF  THE
C          SUBROUTINE  'ESPLINE'  WRITTEN BY  H.P. SUMMERS,   JET   26TH
C          OCTOBER 1989.   IT REMOVES THE COMMON BLOCK  /IONSPL/ ,   THE
C          SWITCHES 'ISW & ISW2' AND ALSO THE CASE FOR THE INTERPOLATION
C          OF CHARGE STATE VALUES.   IT INTRODUCES THE FEATURE  THAT  AN
C          ARRAY OF INPUT  'X-VALUES'  CAN BE  INTERPOLATED/EXTRAPOLATED
C          IN ONE CALL.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          FINTX      ------    EXTERNAL  REAL*8  FUNCTION,   USED  TO
C                               TRANSFORM X-COORDINATES.
C
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     14/01/91 - ADAS91: AS FOR 'XXSPLN' BUT WITH 'LINTRP()' ADDED
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER    NKNOTS       , NIOPT
C-----------------------------------------------------------------------
      PARAMETER( NKNOTS = 101 , NIOPT = 4 )
C-----------------------------------------------------------------------
      INTEGER    IOPT         , NIN       , NOUT      , NOPT
      INTEGER    I            , NIN0      , K         , INTER
```

```
C-----------------------------------------------------------------------
      REAL*8     FINTX
      REAL*8     XK           , XKK
      REAL*8     T1           , T2         , T3       , T4   ,
     &           DL1          , DL2        , DL3
C-----------------------------------------------------------------------
      LOGICAL    LSETX        , LEXTRP
C-----------------------------------------------------------------------
      REAL*8     XIN(NIN)       , YIN(NIN)       ,
     &           XOUT(NOUT)     , YOUT(NOUT)     ,
     &           DY(NIN)
      REAL*8     QVAL(0:NIOPT)  , D2VAL(0:NIOPT) ,
     &           D3VAL(0:NIOPT) , UVAL(0:NIOPT)
      REAL*8     AGRL(4)
      REAL*8     X(NKNOTS)      , DELY(NKNOTS)   ,
     &           H(NKNOTS)      , Q(NKNOTS)      , U(NKNOTS)  ,
     &           D1(NKNOTS)     , D2(NKNOTS)     , D3(NKNOTS)
C-----------------------------------------------------------------------
      LOGICAL    LINTRP(NOUT)   , LUVAL(0:NIOPT)
C-----------------------------------------------------------------------
```

XXSPLN

```
      SUBROUTINE XXSPLN( LSETX , IOPT  , FINTX ,
     &                   NIN   , XIN   , YIN   ,
     &                   NOUT  , XOUT  , YOUT  ,
     &                   DY
     &                 )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXSPLN *********************
C
C  PURPOSE:           TO INTERPOLATE/EXTRAPOLATE USING CUBIC SPLINES
C                     (IF IOPT < 0 NO EXTRAPOLATION TAKES PLACE = VALUES
C                      SET TO ZERO).
C
C  CALLING PROGRAMS:  GENERAL USE
C
C  SUBROUTINE:
C
C  I/O   : (L*4)  LSETX   = .TRUE.  => SET UP SPLINE PARAMETERS RELATING
C                                      TO 'XIN' AXIS.
C                           .FALSE. => DO NOT SET UP SPLINE PARAMETERS
C                                      RELATING TO 'XIN' AXIS.
C                                      (I.E. THEY WERE SET IN A PREVIOUS
C                                            CALL )
C                           ( 'LSETX' IS ALWAYS RETURN AS '.FALSE.'  ON
C                             RETURN FROM THE SUBROUTINE ).
C                           ** IMPORTANT: SEE NOTES BELOW ON 'LSETX' **
C  INPUT : (I*4)  IOPT    = SPLINE END CONDITIONS/EXTRAPOLATION CONTROL
C                           SWITCH - SEE NOTES BELOW
C                           I.E. DEFINES THE BOUNDARY DERIVATIVES.
C                                (VALID VALUES = 0, 1, 2, 3, 4)
C                           IF IOPT < 0 THEN NO EXTRAPOLATION TAKES
C                           - ANY VALUES REQUIRING EXTRAPOLATION WILL BE
C                             SET TO ZERO (END CONDITIONS AS FOR IOPT=0)
C  INPUT : (R*8)  FINTX   = INTERPOLATING X-COORDINATE TRANSFORMATION.
C                           EXTERNAL FUNCTION (SEE ROUTINES BELOW)
C
C  INPUT : (I*4)  NIN     = NUMBER OF KNOTS
C  INPUT : (R*8)  XIN()   = X-VALUES OF KNOTS
C  INPUT : (R*8)  YIN()   = Y-VALUES OF KNOTS
C
C  INPUT : (I*4)  NOUT    = NUMBER OF OUTPUT VALUES TO BE INTERPOLATED
C                           EXTRAPOLATED.
C  INPUT : (R*8)  XOUT()  = X-VALUES AT WHICH INTERPOLATION/EXTRAPOLA-
C                           TION REQUIRED
C  OUTPUT: (R*8)  YOUT()  = INTERPOLATED/EXTRAPOLATED Y-VALUES FOR
C                           REQUESTED 'XOUT()' VALUES.
C
C  OUTPUT: (R*8)  DY()    = INTERPOLATED DERIVATIVES
C
C          (I*4)  NKNOTS  = PARAMETER = MAXIMUM  NUMBER OF KNOTS ALLOWED
C          (I*4)  NIOPT   = PARAMETER = MAXIMUM  VALUE OF IOPT ALLOWED
C
C          (I*4)  I       = GENERAL ARRAY USE
C          (I*4)  K       = INDEX OF 'XOUT()' VALUE FOR INTERPOLATION/
C                           EXTRAPOLATION.
C          (I*4)  NIN0    = 'NIN' - 1
C          (I*4)  INTER   = INDEX OF CLOSEST/NEXT HIGHEST VALUE OF
C                           'XIN()' TO THE VALUE OF 'XOUT()' BEING
C                           INTERPOLATED/EXTRAPOLATED.
C          (I*4)  NOPT    = VALUE OF  'IOPT' USED IN  CALCULATING  END-
C                           CONDITIONS   FOR  STORED  'X-VALUE'  SPLINE
C                           PARAMETERS.   (NOTE:  IF  'IOPT < 0',  THEN
C                           'NOPT = 0'.) - I.E. 'NOPT = MAX( 0, IOPT )'.
C
C          (R*8)  XK      = VALUE OF 'XOUT(K)' BEING INTERPOLATED/
C                           EXTRAPOLATED
C          (R*8)  XKK     = TRANSFORMED VALUE OF 'XOUT(K)' BEING
C                           INTERPOLATED/EXTRAPOLATED.
C          (R*8)  T1      = INVERSE OF SEPARATION OF KNOTS EITHER
C                           SIDE OF CURRENT KNOT.
C          (R*8)  T2      = (CURRENT KNOT POSITION TO NEXT HIGHEST KNOT
```

```
C                          POSITION) DIVIDED BY 'T1'
C          (R*8)  T3      = (CURRENT KNOT POSITION TO NEXT LOWEST  KNOT
C                            POSITION) DIVIDED BY 'T1'
C          (R*8)  T4      = INTERPOLATION FACTOR FOR CURRENT KNOT
C          (R*8)  DL1     = (REQUESTED 'XOUT()' VALUE TO NEXT HIGHEST
C                            KNOT POSITION) DIVIDED BY SEPERATION OF
C                            KNOTS EITHER SIDE OF 'XOUT(K)'.
C          (R*8)  DL2     = (REQUESTED 'XOUT()' VALUE TO NEXT LOWEST
C                            KNOT POSITION) DIVIDED BY SEPERATION OF
C                            KNOTS EITHER SIDE OF 'XOUT(K)'.
C          (R*8)  DL2     = (REQUESTED 'XOUT()' VALUE TO NEXT LOWEST
C          (R*8)  DL3     =  SEPERATION OF KNOTS EITHER SIDE OF
C                            'XOUT(K)' * 'DL1' * 'DL2'.
C
C          (L*4)  LEXTRP  = .TRUE.  => 'EXTRAPOLATION SWITCHED ON'.
C                           .FALSE. => 'EXTRAPOLATION SWITCHED OFF'.
C
C          (R*8)  QVAL()  = VALUE OF 'Q(1)'   : FUNCTION OF 'NOPT'
C          (R*8)  D2VAL() = VALUE OF 'D2(1)'  : FUNCTION OF 'NOPT'
C          (R*8)  D3VAL() = VALUE OF 'D3(1)'  : FUNCTION OF 'NOPT'
C          (R*8)  UVAL() =  VALUE OF 'U(NIN)' : FUNCTION OF 'NOPT'
C          (R*8)  AGRL() =  POLYNOMIAL CONSTANTS FOR CUBIC SPLINE FOR
C                           GIVEN 'XOUT(K)' VALUE.
C          (R*8)  X()     =  TRANSFORMED VALUES OF 'XIN()'
C          (R*8)  H()     =  SEPERATION, ALONG X-AXIS, OF KNOT FROM NEXT
C                            HIGHEST KNOT.
C          (R*8)  Q()     =  SECOND DERIVATIVE FOR KNOT
C          (R*8)  U()     =  TEMPORARY STORAGE OF DECOMPOSED FACTORS
C          (R*8)  DELY() =  SEPERATION, ALONG Y-AXIS, OF KNOT FROM NEXT
C                           HIGHEST KNOT.
C          (R*8)  D1()    =  MULTIPLICATION FACTOR USED IN CALCULATING
C                            'U()'.
C          (R*8)  D2()    =  MULTIPLICATION FACTOR USED IN CALCULATING
C                            'U()'.
C          (R*8)  D3()    =  MULTIPLICATION FACTOR USED IN CALCULATING
C                            'U()'.
C
C          (L*4)  LUVAL()=  .TRUE. => VALUE OF 'UVAL()' REFERS TO RATE
C                                     OF CHANGE OF SLOPE AT FINAL POINT.
C                           .FALSE.=> VALUE OF 'UVAL()' REFERS TO FINAL
C                                     SLOPE
C                            FUNCTION OF 'NOPT'
C
C NOTES: 'LSETX': SET TO .TRUE. ON ENTRY IF A NEW 'XIN' ARRAY IS BEING
C                 USED.  IF THE 'XIN' AXIS IS THE SAME FOR A NUMBER OF
C                 CALLS THEN DO NOT RESET 'LSETX'  -  THIS  SUBROUTINE
C                 SETS IT TO .FALSE. FOR YOU.   IF THE VALUE OF 'NOPT'
C                 IS CHANGED BETWEEN CALLS THEN THE VALUE  OF  'LSETX'
C                 ON  ENTRY IS TAKEN AS BEING EQUAL TO .TRUE. .
C
C                 THEREFORE 'LSETX' NEED ONLY BE SET TO .TRUE. ON ENTRY
C                 IF EITHER IT IS ITS FIRST CALL OR IF ANY ONE  OF  THE
C                 FOLLOWING VALUES HAS CHANGED:
C
C                 'NIN' , 'FINTX' , 'XIN(I), I=1,NIN'
C
C                 CARE: A VARIABLE MUST BE USED FOR 'LSETX', A CONSTANT,
C                       I.E.  .TRUE. ,  CANNOT BE DIRECTLY TYPED AS  AN
C                       ARGUMENT BECAUSE IT WILL BE CHANGED TO  .FALSE.
C                       ON RETURN.
C
C         SPLINE  END CONDITIONS AND EXTRAPOLATION DEPEND ON 'IOPT' AS
C         FOLLOWS:
C
C         ------------------------------------------------------------
C         | IOPT  | NOPT |  DY(1)  DDY(1)  |  DY(N)   DDY(N)  |EXTRAP'N|
C         |-------|------|-----------------|------------------|--------|
C         | < 0   |   0  |    -     0.0    |    -      0.0    |  NO    |
C         |   0   |   0  |    -     0.0    |    -      0.0    |  YES   |
C         |   1   |   1  |    -     0.0    |  -1.5      -     |  YES   |
C         |   2   |   2  |   0.0     -     |   1.0      -     |  YES   |
C         |   3   |   3  |  -0.5     -     |  -1.5      -     |  YES   |
C         |   4   |   4  |   0.0     -     |    -      0.0    |  YES   |
C         ------------------------------------------------------------
C
C            NB. OPTIONS TO BE EXTENDED FOR POWER AND CX APPLICATION
C
C         ------------------------------------------------------------
C          IF ( IOPT.LT.0 ) - NO EXTRAPOLATION TAKES PLACE VALUES SET
C                             TO ZERO (CARE IF LOG OF OUTPUT IS NEEDED).
C          IF ( IOPT.GT.4 ) PROGRAM STOPS
C         ------------------------------------------------------------
C
C          THIS SUBROUTINE IS AN AMENDED  AND STRUCTURED VERSION OF  THE
C          SUBROUTINE  'ESPLINE'  WRITTEN BY  H.P. SUMMERS,   JET   26TH
C          SUBROUTINE  'ESPLINE'  WRITTEN BY  H.P. SUMMERS,   JET   26TH
C          OCTOBER 1989.   IT REMOVES THE COMMON BLOCK  /IONSPL/ ,   THE
C          SWITCHES 'ISW & ISW2' AND ALSO THE CASE FOR THE INTERPOLATION
C          OF CHARGE STATE VALUES.   IT INTRODUCES THE FEATURE  THAT  AN
C          ARRAY OF INPUT  'X-VALUES'  CAN BE  INTERPOLATED/EXTRAPOLATED
C          IN ONE CALL.
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          FINTX      ------    EXTERNAL  REAL*8  FUNCTION,   USED  TO
C                               TRANSFORM X-COORDINATES.
C
```

```
C
C AUTHOR:   PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C           K1/0/81
C           JET EXT. 4569
C
C DATE:     10/08/90 (30/08/90: IOPT = 4 ADDED & 'LUVAL' PARAMETER)
C
C UPDATE:   17/01/91 - PE BRIDEN: ADAS91 - IOPT < 0 ADDED - NO EXTRAP'N.
C                                        - NOPT DEFINITION CHANGED.
C                                        - INTRODUCED 'LEXTRP'.
C UNIX-IDL PORT:
C
C VERSION: 1.1                          DATE: 08-11-95
C MODIFIED: TIM HAMMOND (TESSELLA SUPPORT SERVICES PLC)
C               - FIRST RELEASE
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER    NKNOTS       , NIOPT
C-----------------------------------------------------------------------
      PARAMETER( NKNOTS = 101 , NIOPT = 4 )
C-----------------------------------------------------------------------
      INTEGER    IOPT         , NIN       , NOUT      , NOPT
      INTEGER    I            , NIN0      , K         , INTER
C-----------------------------------------------------------------------
      REAL*8     FINTX
      REAL*8     XK           , XKK
      REAL*8     T1           , T2         , T3       , T4   ,
     &           DL1          , DL2        , DL3
C-----------------------------------------------------------------------
      LOGICAL    LSETX        , LEXTRP
C-----------------------------------------------------------------------
      REAL*8     XIN(NIN)       , YIN(NIN)       ,
     &           XOUT(NOUT)     , YOUT(NOUT)     ,
     &           DY(NIN)
      REAL*8     QVAL(0:NIOPT)  , D2VAL(0:NIOPT) ,
     &           D3VAL(0:NIOPT) , UVAL(0:NIOPT)
      REAL*8     AGRL(4)
      REAL*8     X(NKNOTS)      , DELY(NKNOTS)   ,
     &           H(NKNOTS)      , Q(NKNOTS)      , U(NKNOTS)  ,
     &           D1(NKNOTS)     , D2(NKNOTS)     , D3(NKNOTS)
C-----------------------------------------------------------------------
      LOGICAL    LUVAL(0:NIOPT)
C-----------------------------------------------------------------------
```

XXSPZD

```
      SUBROUTINE XXSPZD( USRGRP , USRTYP , USREXT )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 SUBROUTINE: XXSPZD *********************
C
C  PURPOSE: ADAS ROUTINE - SETS UP THE DEFAULT USEGRP, USRTYP AND USREXT
C           WHICH IDENTIFY THE FILENAME AND EXTENSION TO BE READ IN
C           SUBROUTINE SPZD.  IT WORKS IN THE SAME MANNER AS XXUID WHICH
C           WHICH ALLOWS THE DEFAULT USER SPACE TO BE SET
C
C
C           USRGRP: VALUE ON INPUT  =>  USRGRP: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA GROUPNAME
C                     *                 DEFAULT ADAS DATA GROUPNAME
C                  <BLANK>              *** USRGRP VALUE NOT CHANGED ***
C                  <OTHER>              *** USRGRP VALUE NOT CHANGED ***
C
C
C           USRTYP: VALUE ON INPUT  =>  USRTYP: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA TYPENAME
C                     *                 DEFAULT ADAS DATA TYPENAME
C                  <BLANK>              *** USRTYP VALUE NOT CHANGED ***
C                  <OTHER>              *** USRTYP VALUE NOT CHANGED ***
C
C
C           USREXT: VALUE ON INPUT  =>  USREXT: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA MEMBER EXTENS.
C                     *                 DEFAULT ADAS DATA MEMBER EXTENS
C                  <BLANK>              *** USREXT VALUE NOT CHANGED ***
C                  <OTHER>              *** USREXT VALUE NOT CHANGED ***
C
C         ? => QUERIES CURRENT ADAS DATA USRGRP, USRTYP OR USREXT
C              SETTING.
C         * => SETS ADAS DATA USEGRP, USRTYP OR USREXT SETTING
C              TO DEFAULT VALUE.
C   <BLANK> => SETS ADAS DATA USRGRP, USRTYP OR USREXT SETTING
C              TO DEFAULT VALUE.
C   <OTHER> => SETS ADAS DATA USRGRP, USRTYP OR USREXT SETTING
C              TO INPUT   VALUE.
C
C  CALLING PROGRAM: SPZD AND MAIN PROGRAMS USING SPZD
C
C  SUBROUTINE:
```

```
C
C  I/O   : (C*8)  USRGRP   = USRGRP UNDER WHICH ADAS DATA IS STORED
C                            (IF BLANK DEFAULTS TO DEFGRP)
C
CA I/O   : (C*80) USRTYP   = UNDER UNIX THIS IS OPTIONAL SUB-DIRECTORY
C                            (IF BLANK DEFAULTS TO DEFTYP)
C
C  I/O   : (C*3)  USREXT   = USREXT UNDER WHICH ADAS DATA IS STORED
C                            (IF BLANK DEFAULTS TO DEFEXT)
C
C          (C*8)  DEFGRP   = PARAMETER = DEFAULT USER GROUP FOR ADAS
C                                        DATA SOURCE
C
CA         (C*80) DEFTYP   = PARAMETER = DEFAULT SUB-DIRECTORY NAME
C                                        WHICH IS BLANK.
C
C          (C*3)  DEFEXT   = PARAMETER = DEFAULT USER EXTENSION FOR ADAS
C                                        DATA SOURCE
C
C          (C*8)  ADASGR   = CURRENT ADAS DATA SOURCE GROUP
C
CA         (C*80) ADASTY   = CURRENT SUBDIRECTORY NAME
C
C          (C*3)  ADASEX   = CURRENT ADAS DATA SOURCE EXTENSION
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C NOTE:
C          TO CHECK CURRENT ADAS SOURCE USRGRP, USRTYP AND USREXT
C          CALL XXSPZD WITH ?`S AS INPUTS.
C
C AUTHOR:  HUGH P. SUMMERS, JET
C          K1/1/67
C          JET EXT. 4941
C
C DATE:     2/09/93
C
C UPDATE:  21/11/94 - L. JALOTA - ALTERED CHARACTER STRING LENGTHS FOR UNIX
C
C-----------------------------------------------------------------------
      CHARACTER  DEFGRP*8             , DEFTYP*80     , DEFEXT*3
C-----------------------------------------------------------------------
      PARAMETER  (DEFGRP = 'ionelec ' , DEFTYP = ' ' ,
     &            DEFEXT = 'pzd')
C-----------------------------------------------------------------------
      CHARACTER  USRGRP*8             , ADASGR*8      ,
     &           USRTYP*80            , ADASTY*80     ,
     &           USREXT*3             , ADASEX*3
C-----------------------------------------------------------------------
```

XXSSXB

```
      SUBROUTINE XXSSXB( USRGRP , USRTYP , USREXT )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXSSXB *********************
C
C  PURPOSE: ADAS ROUTINE - SETS UP THE DEFAULT USRGRP, USRTYP AND USREXT
C           WHICH IDENTIFY THE FILENAME AND EXTENSION TO BE READ IN
C           SUBROUTINE SSXB.  IT WORKS IN THE SAME MANNER AS XXUID WHICH
C           WHICH ALLOWS THE DEFAULT USER SPACE TO BE SET
C
C
C           USRGRP: VALUE ON INPUT  =>  USRGRP: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA GROUPNAME
C                     *                 DEFAULT ADAS DATA GROUPNAME
C                   <BLANK>             *** USRGRP VALUE NOT CHANGED ***
C                   <OTHER>             *** USRGRP VALUE NOT CHANGED ***
C
C
C           USRTYP: VALUE ON INPUT  =>  USRTYP: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA TYPENAME
C                     *                 DEFAULT ADAS DATA TYPENAME
C                   <BLANK>             *** USRTYP VALUE NOT CHANGED ***
C                   <OTHER>             *** USRTYP VALUE NOT CHANGED ***
C
C
C           USREXT: VALUE ON INPUT  =>  USREXT: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA MEMBER EXTENS.
C                     *                 DEFAULT ADAS DATA MEMBER EXTENS
C                   <BLANK>             *** USREXT VALUE NOT CHANGED ***
C                   <OTHER>             *** USREXT VALUE NOT CHANGED ***
C
C         ? => QUERIES CURRENT ADAS DATA USRGRP, USRTYP OR USREXT
C              SETTING.
C         * => SETS ADAS DATA USRGRP USRTYP OR USREXT SETTING
C              TO DEFAULT VALUE.
C   <BLANK> => SETS ADAS DATA USRGRP, USRTYP OR USREXT SETTING
```

```
C               TO DEFAULT VALUE.
C    <OTHER> => SETS ADAS DATA USRGRP, USRTYP OR USREXT SETTING
C               TO INPUT   VALUE.
C
C  CALLING PROGRAM: SSXB AND MAIN PROGRAMS USING SSXB
C
C  SUBROUTINE:
C
CA I/O   : (C*8)  USRGRP   = USRGRP UNDER WHICH ADAS DATA IS STORED
C                            (IF BLANK DEFAULTS TO DEFGRP)
C
CA I/O   : (C*80  USRTYP   = OPTIONAL SUB-DIRECTORY UNDER UNIX
C                            (IF BLANK DEFAULTS TO DEFTYP)
C
C  I/O   : (C*3)  USREXT   = USREXT UNDER WHICH ADAS DATA IS STORED
C                            (IF BLANK DEFAULTS TO DEFEXT)
C
CA         (C*8)  DEFGRP   = PARAMETER = DEFAULT USER GROUP FOR ADAS
C                                        DATA SOURCE
C
CA         (C*80) DEFTYP   = PARAMETER = DEFAULT USER TYPE FOR ADAS
C                                        DATA SOURCE
C
C          (C*3)  DEFEXT   = PARAMETER = DEFAULT USER EXTENSION FOR ADAS
C                                        DATA SOURCE
C
CA         (C*8)   ADASGR   = CURRENT ADAS DATA SOURCE GROUP
C
CA         (C*80)  ADASTY   = CURRENT SUB-DIRECTORY NAME ( OPTIONAL)
C
C          (C*3)  ADASEX   = CURRENT ADAS DATA SOURCE EXTENSION
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C NOTE:
C          TO CHECK CURRENT ADAS SOURCE USRGRP, USRTYP AND USREXT
C          CALL XXSSXB WITH '?' AS INPUTS.
C
C AUTHOR:  HUGH P. SUMMERS, JET
C          K1/1/57
C          JET EXT. 4941
C
C DATE:     2/09/93
C
C UPDATES: 24/10/94  L. JALOTA (TESSELLA SUPPORT SERVICES PLC)
C                     CHANGED VALUES OF DEFGRP,DEFTYP, DEFEXT SUITABLE
C                     FOR DEC ALPHA DIRECTORY STRUCTURE.
C UPDATE : 22/11/94  L. JALOTA  -  TIDIED UP CHARACTER LENGTH DEFINITIONS.
C
C-----------------------------------------------------------------------
      CHARACTER  DEFGRP*8        , DEFTYP*80        , DEFEXT*3
C-----------------------------------------------------------------------
      PARAMETER  (DEFGRP = 'ionelec ' , DEFTYP = ' ' ,
     &            DEFEXT = 'sxb')
C-----------------------------------------------------------------------
      CHARACTER  USRGRP*8             , ADASGR*8      ,
     &           USRTYP*80              , ADASTY*80     ,
     &           USREXT*3             , ADASEX*3
C-----------------------------------------------------------------------
```

XXSSZD

```
      SUBROUTINE XXSSZD( USRGRP , USRTYP , USREXT )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ***************** FORTRAN77 SUBROUTINE: XXSSZD *********************
C
C  PURPOSE: ADAS ROUTINE - SETS UP THE DEFAULT USEGRP, USRTYP AND USREXT
C           WHICH IDENTIFY THE FILENAME AND EXTENSION TO BE READ IN
C           SUBROUTINE SSZD.  IT WORKS IN THE SAME MANNER AS XXUID WHICH
C           WHICH ALLOWS THE DEFAULT USER SPACE TO BE SET
C
C
C           USRGRP: VALUE ON INPUT  =>  USRGRP: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA GROUPNAME
C                     *                 DEFAULT ADAS DATA GROUPNAME
C                   <BLANK>             *** USRGRP VALUE NOT CHANGED ***
C                   <OTHER>             *** USRGRP VALUE NOT CHANGED ***
C
C
C           USRTYP: VALUE ON INPUT  =>  USRTYP: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA TYPEPNAME
C                     *                 DEFAULT ADAS DATA TYPEGROUPNAME
C                   <BLANK>             *** USRTYP VALUE NOT CHANGED ***
C                   <OTHER>             *** USRTYP VALUE NOT CHANGED ***
C
C
C           USREXT: VALUE ON INPUT  =>  USREXT: VALUE ON OUTPUT
C
```

```
C                     ?                CURRENT ADAS DATA MEMBER EXTENS.
C                     *                DEFAULT ADAS DATA MEMBER EXTENS
C                  <BLANK>             *** USREXT VALUE NOT CHANGED ***
C                  <OTHER>             *** USREXT VALUE NOT CHANGED ***
C
C         ? => QUERIES CURRENT ADAS DATA USRGRP, USRTYP OR USREXT
C              SETTING.
C         * => SETS ADAS DATA USEGRP, USRTYP OR USREXT SETTING
C              TO DEFAULT VALUE.
C   <BLANK> => SETS ADAS DATA USRGRP, USRTYP OR USREXT SETTING
C              TO DEFAULT VALUE.
C   <OTHER> => SETS ADAS DATA USRGRP, USRTYP OR USREXT SETTING
C              TO INPUT   VALUE.
C
C  CALLING PROGRAM: SSZD AND MAIN PROGRAMS USING SSZD
C
C  SUBROUTINE:
C
C I/O   : (C*8)  USRGRP   = USRGRP UNDER WHICH ADAS DATA IS STORED
C                           (IF BLANK DEFAULTS TO DEFGRP)
C
CX I/O   : (C*6)  USRTYP   = USRTYP UNDER WHICH ADAS DATA IS STORED
CA I/O   : (C*80) USRTYP   = OPTIONAL SUBDIRECTORY FOR FILE
C                           (IF BLANK DEFAULTS TO DEFTYP)
C
CX I/O   : (C*3)  USREXT   = USREXT UNDER WHICH ADAS DATA IS STORED
C                           (IF BLANK DEFAULTS TO DEFEXT)
C
C          (C*8)  DEFGRP   = PARAMETER = DEFAULT USER GROUP FOR ADAS
C                                        DATA SOURCE
C
CX         (C*6)  DEFTYP   = PARAMETER = DEFAULT USER TYPE FOR ADAS
CA         (C*80) DEFTYP   = OPTIONAL SUB-DIRECTORY DEFAULT IS BLANK
C                                        DATA SOURCE
C
C          (C*3)  DEFEXT   = PARAMETER = DEFAULT USER EXTENSION FOR ADAS
C                                        DATA SOURCE
C
C          (C*8)  ADASGR   = CURRENT ADAS DATA SOURCE GROUP
C
CX         (C*6)  ADASTY   = CURRENT ADAS DATA SOURCE TYPE
CA         (C*80) ADASTY   = OPTIONAL SUB-DIRECTORY
C
C          (C*3)  ADASEX   = CURRENT ADAS DATA SOURCE EXTENSION
C
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C NOTE:
C          TO CHECK CURRENT ADAS SOURCE USRGRP, USRTYP AND USREXT
C          CALL XXSSZD WITH ?`S AS INPUTS.
C
C AUTHOR:  HUGH P. SUMMERS, JET
C          K1/1/67
C          JET EXT. 4941
C
C DATE:     2/09/93
C
C UPDATE:  L. JALOTA - 10/11/94 : MODIFIED FOR USE UNDER UNIX
C                                 INCREASED STRING LENGTHS
C UPDATE:  L. JALOTA - 22/11/94 : TIDIED UP CHARACTER LENGTH DEFINITIONS.
C-----------------------------------------------------------------------
CX    CHARACTER  DEFGRP*8             , DEFTYP*6      , DEFEXT*3
      CHARACTER  DEFGRP*8             , DEFTYP*80     , DEFEXT*3
C-----------------------------------------------------------------------
CX    PARAMETER  (DEFGRP = 'IONELEC ' , DEFTYP = 'DATA ' ,
CX   &            DEFEXT = 'SZD')
      PARAMETER  (DEFGRP = 'ionelec'  , DEFTYP = ' ' ,
     &            DEFEXT = 'szd')
C-----------------------------------------------------------------------
CX    CHARACTER  USRGRP*8             , ADASGR*8      ,
CX   &           USRTYP*6             , ADASTY*6      ,
      CHARACTER  USRGRP*8             , ADASGR*8      ,
     &           USRTYP*80            , ADASTY*80      ,
     &           USREXT*3             , ADASEX*3
C-----------------------------------------------------------------------
```

XXSTNP

```
      SUBROUTINE XXSTNP( IUNIT , PGLEN  , NBLOCK , NLINES )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ******************* FORTRAN77 SUBROUTINE: XXSTNP ******************
C
C  PURPOSE: CHECK WHETHER OUTPUT BLOCK WILL FIT ON CURRENT PAGE. IF NOT
C           A NEW PAGE IS INITIATED, AND LINE COUNT RESET ACCORDINGLY.
C
C  CALLING PROGRAM: GENERAL USE
C
C
C  SUBROUTINE:
C
```

```
C  INPUT : (I*4)  IUNIT   = OUTPUT UNIT FOR RESULTS
C  INPUT : (I*4)  PGLEN   = OUTPUT PAGE LENGTH IN LINES
C  INPUT : (I*4)  NBLOCK  = NUMBER OF LINES IN BLOCK
C  I/O   : (I*4)  NLINES  = LAST PAGE LINE WRITTEN TOO.
C
C ROUTINES: NONE
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    05/07/90
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER  IUNIT , PGLEN , NBLOCK , NLINES
C-----------------------------------------------------------------------
```

XXSTUC

```
      SUBROUTINE XXSTUC( STRING )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXSTUC *********************
C
C  PURPOSE: ENSURES ALL LETTERS IN INPUT STRING ARE UPPER CASE.
C
C  CALLING PROGRAM: GENERAL USE.
C
C  I/O   : (C*(*)) STRING = STRING.
C
C          (I*4)  IDIFF   = DIFFERENCE IN CODES BETWEEN LOWER AND UPPER
C                           CASE LETTER.
C          (I*4)  I       = LOOP INDEX.
C
C ROUTINES: NONE
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    26/11/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      INTEGER    IDIFF   , I
C-----------------------------------------------------------------------
      CHARACTER  STRING*(*)
C-----------------------------------------------------------------------
```

XXTCON

```
      SUBROUTINE XXTCON( INTYP, OUTTYP, IZ1, ITVAL, TIN, TOUT )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXTCON *********************
C
C  PURPOSE: TO CONVERT AN ARRAY OF TEMPERATURES INTO SPECIFIED UNITS
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT :    (I*4)  INTYP   = 1 => 'TIN(array)' UNITS: KELVIN
C                              = 2 => 'TIN(array)' UNITS: eV
C                              = 3 => 'TIN(array)' UNITS: REDUCED TEMP.
C  INPUT :    (I*4)  OUTTYP  = 1 => 'TOUT(array)' UNITS: KELVIN
C                              = 2 => 'TOUT(array)' UNITS: eV
C                              = 3 => 'TOUT(array)' UNITS: REDUCED TEMP.
C  INPUT :    (I*4)  IZ1     = RECOMBINING ION CHARGE (= Z+1).
C  INPUT :    (I*4)  ITVAL   = NUMBER OF TEMPERATURES IN 'TIN(array)'
C  INPUT :    (R*8)  TIN()   = INPUT TEMPERATURES (STATED UNITS)
C  OUTPUT:    (R*8)  TOUT()  = OUTPUT TEMPERATURES (STATED UNITS)
C
C             (R*8)  EV2KEL  = ELECTRON VOLTS TO KELVIN CONVERSION
C             (R*8)  KEL2EV  = KELVIN TO ELECTRON VOLTS CONVERSION
C
C             (I*4)  I       = GENERAL USE
C
C             (R*8)  Z1P2    = 'IZ1'**2
C             (R*8)  TCONV() = TEMPERATURE CONVERSION PARAMETERS
C
C ROUTINES:  NONE
C
C NOTE:
C            TEMPERATURE CONVERSION PARAMETERS:
C
C            INTYP = 1 ; TCONV(1) =>             KELVIN  -> OUTPUT UNITS
```

```
C            INTYP = 2 ; TCONV(2) =>                eV  -> OUTPUT UNITS
C            INTYP = 3 ; TCONV(3) => REDUCED TEMPERATURE -> OUTPUT UNITS
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    04/01/91
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      REAL*8     EV2KEL             , KEL2EV
C-----------------------------------------------------------------------
      PARAMETER( EV2KEL=1.16054D+04 , KEL2EV=8.61668D-05 )
C-----------------------------------------------------------------------
      INTEGER    INTYP              , OUTTYP            ,
     &           IZ1                , ITVAL
      INTEGER    I
C-----------------------------------------------------------------------
      REAL*8     TIN(ITVAL)         , TOUT(ITVAL)
      REAL*8     Z1P2               , TCONV(3)
C-----------------------------------------------------------------------
```

XXTERM

```
      SUBROUTINE XXTERM
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXTERM *********************
C
C  PURPOSE: TERMINATES PROGRAM WITH MESSAGE.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE: NO VARIABLES
C
C ROUTINES: NONE
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 4569
C
C DATE:    17/08/90
C
C-----------------------------------------------------------------------
```

XXUID

```
      SUBROUTINE XXUID( USERID )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXUID  *********************
C
C  PURPOSE: ADAS ROUTINE - SETS UP THE DEFAULT USERID WHICH STORES THE
C           DATA TO BE READ USING STANDARD ADAS DATA READING ROUTINES.
CA           UNDER UNIX PORT THIS NOW GETS THE ENVIRONMENT VARIABLE
CA           "ADASCENT".
C
C           USERID: VALUE ON INPUT  =>  USERID: VALUE ON OUTPUT
C
C                     ?                 CURRENT ADAS DATA SOURCE USER ID
C                     *                 DEFAULT ADAS DATA SOURCE USER ID
C                   <BLANK>             *** USERID VALUE NOT CHANGED ***
C                   <OTHER>             *** USERID VALUE NOT CHANGED ***
C
C           ? => QUERIES CURRENT ADAS SOURCE USERID SETTING.
C           * => SETS ADAS SOURCE USERID SETTING TO DEFAULT VALUE.
C     <BLANK> => SETS ADAS SOURCE USERID SETTING TO DEFAULT VALUE.
C     <OTHER> => SETS ADAS SOURCE USERID SETTING TO INPUT   VALUE.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
CX  I/O   : (C*6)  USERID   = USERID UNDER WHICH ADAS DATA IS STORED
CX                            (IF BLANK DEFAULTS TO DEFUID)
CX
CX         (C*6)  DEFUID   = PARAMETER = DEFAULT USER ID FOR ADAS DATA
CX                                       SOURCE
CX
CX         (C*6)  ADASID   = CURRENT ADAS DATA SOURCE USER ID
C
C  I/O   : (C*80)  USERID   = USERID UNDER WHICH ADAS DATA IS STORED
C                             (IF BLANK DEFAULTS TO DEFUID)
C
C          (C*80)  DEFUID   = PARAMETER = DEFAULT USER ID FOR ADAS DATA
C                                         SOURCE
C
C          (C*80)  ADASID   = CURRENT ADAS DATA SOURCE USER ID
```

```
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C
C NOTE:
C          TO CHECK CURRENT ADAS SOURCE USERID CALL XXUID WITH
C          ? AS INPUT.
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 5023
C
C DATE:    10/03/93
C
C UPDATE:  L. JALOTA  - 7/11/94 : ADDED CALL TO GETENV TO FETCH UNIX
C                                 ENVIRONMENT VARIABLE ADASUSER
C
C-----------------------------------------------------------------------
      CHARACTER  DEFUID*80
C-----------------------------------------------------------------------
      CHARACTER  USERID*80          , ADASID*80              , ENV*80
C-----------------------------------------------------------------------
      EXTERNAL   GETENV
C-----------------------------------------------------------------------
```

XXWORD

```
      SUBROUTINE XXWORD( CTEXT  , CDELIM , NFIRST ,
     &                   IWORDS ,
     &                   IFIRST , ILAST  , NWORDS
     &                 )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ****************** FORTRAN77 SUBROUTINE: XXWORD *********************
C
C  PURPOSE: TO EXTRACT THE Nfirst to (Nfirst+IWORDS-1) WORDS FROM AN
C           INPUT STRING. OUTPUTS THE FIRST AND LAST BYTE INDEXES OF
C           EACH WORD AS WELL AS THE TOTAL NUMBER OF WORDS FOUND.
C
C           A WORD = A STRING OF CHARACTERS SEPARATED BY ANY CHARACTER
C                    CONTAINED IN THE INPUT STRING CDELIM.
C
C  CALLING PROGRAM: GENERAL USE
C
C  SUBROUTINE:
C
C  INPUT : (C*(*)) CTEXT   = INPUT TEXT LINE CONTAINING STRING
C  INPUT : (C*(*)) CDELIM  = INPUT STRING CONTAINING DELIMITER CHARS.
C  INPUT : (I*4)   NFIRST  = THE INDEX NO. OF THE FIRST WORD TO EXTRACT.
C
C  I/O   : (I*4)   IWORDS  = INPUT : SIZE OF IFIRST, ILAST(ARRAYS)
C                                    (I.E. NUMBER OF WORDS TO EXTRACT)
C                            = OUTPUT: NUMBER OF REQUESTED WORDS FOUND
C
C  OUTPUT: (I*4)   IFIRST()= INDEX OF FIRST BYTE OF THE Nth WORD
C  OUTPUT: (I*4)   ILAST() = INDEX OF LAST  BYTE OF THE Nth WORD
C  OUTPUT: (I*4)   NWORDS  = THE TOTAL NUMBER OF WORDS FOUND IN CTEXT
C
C           (I*4)   LENTXT  = LENGTH IN BYTES OF 'CTEXT' STRING
C           (I*4)   IDELIM  = 0 => CTEXT CHARACTER IS NOT A DELIMITER
C                             > 0 => CTEXT CHARACTER IS A DELIMITER
C           (I*4)   ITOTAL  = NUMBER OF WORDS FOUND SO FAR
C           (I*4)   IINDEX  = IFIRST()/ILAST() INDEX OF CURRENT WORD
C           (I*4)   NLAST   = THE INDEX NO. OF THE LAST WORD TO EXTRACT
C           (I*4)   I       = GENERAL USE INDEX
C
C           (L*4)   LWORD   = .TRUE.  - PROCESSING AN IDENTIFIED WORD
C                             .FALSE. - PROCESSING SPACE BETWEEN WORDS
C
C ROUTINES: NONE
C
C NOTES:    IF THERE IS NO Nfirst WORD OR NO WORDS ARE FOUND
C           (I.E. INPUT STRING IS BLANK) THEN IWORDS=0
C
C AUTHOR:  PAUL E. BRIDEN (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/37
C          JET EXT. 5023
C
C DATE:    20/05/93
C
C-----------------------------------------------------------------------
      INTEGER    NFIRST          , IWORDS          , NWORDS      ,
     &           LENTXT          , IDELIM          , ITOTAL      ,
     &           IINDEX          , NLAST           , I
C-----------------------------------------------------------------------
      LOGICAL    LWORD
C-----------------------------------------------------------------------
      CHARACTER  CTEXT*(*)       , CDELIM*(*)
C-----------------------------------------------------------------------
      INTEGER    IFIRST(IWORDS)  , ILAST(IWORDS)
C-----------------------------------------------------------------------
```

XXYCF2

```
      SUBROUTINE XXYCF2(FCN, M, N, X, FVEC, FJAC, B, C)
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C ROUTINE: XXYCF2
C
C PURPOSE: CALCULATES AN ESTIMATE OF THE ELEMENTS OF THE VARIANCE-COVARIANCE
C          MATRIX OF THE ESTIMATED REGRESSION COEFFICIENTS FOR A NON-LINEAR
C          LEAST-SQUARES PROBLEM. REPLACES NAG ROUTINE E04YCF WHEN CALLED
C          AFTER E04GCF OR ANOTHER ROUTINE THAT USES FIRST DERIVATIVES.
C***********************************************************************
C          THIS ROUTINE CAN ONLY BE USED WHEN FIRST DERIVATIVES ARE AVAILABLE.
C***********************************************************************
C          SEE THE DOCUMENTATION OF E04YCF AND THE E04 SERIES FOR MORE
C          INFORMATION ON THE MATHEMATICS OF CALCULATING THE VARIANCE-
C          COVARIANCE MATRIX.
C
C CALLING PROGRAM: GENERAL USE
C
C INPUT:
C    SUBROUTINE FCN - CALCULATES THE FUNCTIONS AND THEIR DERIVATIVES.
C                     IF ONE IS DIRECTLY REPLACING THE NAG ROUTINE THEN
C                     THIS ARGUMENT WILL BE LSFUN2 IN THE CALLING
C                     PROGRAM, AND LSFUN2 WILL HAVE TO BE MODIFIED TO
C                     ACCOMMODATE THE IFLAG VARIABLE. SEE BELOW FOR DETAILS.
C    (I*4)  M       - NUMBER OF FUNCTIONS
C    (I*4)  N       - NUMBER OF VARIABLES, N<=M
C    (R*8)  X       - VECTOR OF THE ESTIMATED SOLUTION
C    (R*8)  FVEC()  - THE FUNCTION EVALUATED AT X
C    (R*8)  FJAC()  - A WORK MATRIX OF DIMENSION (M,N)
C    (R*8)  B()     - A WORK VECTOR OF DIMENSION N
C
C OUTPUT:
C     (R*8)  C(,)     - THE VARIANCE-COVARIANCE MATRIX. DIMENSIONS NxN.
C
C ROUTINES:
C-----------------------------------------------------------------------
C     NAME     SOURCE   PURPOSE
C-----------------------------------------------------------------------
C     XXMINV   ADAS     INVERTS A SQUARE MATRIX
C     FCN      USER     CALCULATES FUNCTIONS AND THEIR FIRST DERIVATIVES
C {
C    SUBROUTINE FCN(M,N,X,FVEC,FJAC,LDFJAC,IFLAG)
C    INTEGER M,N,IFLAG, LDFJAC
C    REAL*8 X(N), FVEC(M), FJAC(LDFJAC,N)
C  INPUT:
C    M,N    - AS ABOVE
C    LDFJAC - FIRST DIMENSION OF FJAC
C    X      - VECTOR AT WHICH THE FUNCTION IS TO BE EVALUATED
C    IFLAG  - 1=> CALCULATE FVEC BUT DON'T CHANGE FJAC
C           - 2=> CALCULATE FJAC BUT DON'T CHANGE FVEC
C  OUTPUT:
C    FVEC   - FUNCTION EVALUATED AT X OR NOT CHANGED
C    FJAC   - JACOBIAN OF FUNCTION EVALUATED AT X OR NOT CHANGED
C }
C-----------------------------------------------------------------------
C
C AUTHOR: WILLIAM OSBORN (TESSELLA SUPPORT SERVICES PLC.)
C
C DATE:   31-05-96
C
C VERSION 1.1                           DATE: 31-05-96
C MODIFIED: WILLIAM OSBORN
C           - FIRST VERSION
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER M, N, I, J, IFLAG, K, LWA
      REAL*8 X(N), C(N,N), FSUMSQ, FJAC(M,N), FVEC(M), B(N)
      REAL*8 FAC, DINT
      LOGICAL LSOLVE
      EXTERNAL FCN
```

Z16CD

```
      FUNCTION Z16CD( IB  , IV  , IO1 , IO2 , IO3 ,
     &                IT  , IG  , IS  ,
     &                QN  , PM  , ZT1 , ETA , NA  , LA
     &          )
      IMPLICIT NONE
C-----------------------------------------------------------------------
C
C  ************** FORTRAN77 COMPLEX*16 FUNCTION: Z16CD  ***************
C
C  PURPOSE:
C
C  CALLING PROGRAM: GENERAL USE
C
C  FUNC:   (R*8)  Z16CD   =
C
C  INPUT:  (I*4)  IB      =
```

```
C  INPUT:  (I*4)  IV      =
C  INPUT:  (I*4)  IO1     =
C  INPUT:  (I*4)  IO2     =
C  INPUT:  (I*4)  IO3     =
C  INPUT:  (I*4)  IT      =
C  INPUT:  (I*4)  IG      =
C  INPUT:  (I*4)  IS      =
C  INPUT:  (R*8)  QN      =
C  INPUT:  (R*8)  PM      =
C  INPUT:  (R*8)  ZT1     = EFFECTIVE CHARGE FOR THE 1S ELECTRON OF THE
C                           TARGET ATOM IN THE FINAL STATE.
C  INPUT:  (R*8)  ETA     =
C  INPUT:  (I*4)  NA      = N QUANTUM NUMBER OF THE INITIAL STATE.
C  INPUT:  (I*4)  LA      = L QUANTUM NUMBER OF THE INITIAL STATE.
C
C          (I*4)  JB      =
C          (I*4)  I       = LOOP INDEX.
C
C          (R*8)  XNA     = REAL VALUE = NA.
C          (R*8)  XS      = REAL VALUE = S.
C          (R*8)  T1      =
C          (R*8)  T2      =
C          (R*8)  T3      =
C
C          (Z*16) CT1     =
C          (Z*16) CT2     =
C
C ROUTINES:
C          ROUTINE    SOURCE    BRIEF DESCRIPTION
C          ------------------------------------------------------------
C          I4JGAM     ADAS      RETURNS VALUE FROM 'JGAM' TABLE.
C          R8GAM      ADAS      RETURNS VALUE FROM 'GAM' TABLE.
C
C AUTHOR:  JONATHAN NASH (TESSELLA SUPPORT SERVICES PLC)
C          K1/0/81
C          JET EXT. 5183
C
C DATE:    05/10/93
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      COMPLEX*16  Z16CD
C-----------------------------------------------------------------------
      INTEGER     I4JGAM
      REAL*8      R8GAM
C-----------------------------------------------------------------------
      INTEGER     IB      , IV      , IO1     , IO2     , IO3     ,
     &            IT      , IG      , IS      , NA      , LA
      INTEGER     JB      , I
C-----------------------------------------------------------------------
      REAL*8      QN      , PM      , ZT1     , ETA
      REAL*8      XNA     , XS      , T1      , T2      , T3
C-----------------------------------------------------------------------
      COMPLEX*16  CT1     , CT2
C-----------------------------------------------------------------------
```