KfK 4454 B
September 1988

IMFREM
A Computer Program for Analyses of Intermediate Mass-Fragment Emission in Heavy Ion Reactions

I. M. Brâncuş, J. Oehlschläger, J. Wentz
Institut für Kernphysik

Kernforschungszentrum Karlsruhe

KERNFORSCHUNGSZENTRUM KARLSRUHE

Institut für Kernphysik

KfK 4454 B

IMFREM : A COMPUTER PROGRAM FOR ANALYSES OF INTERMEDIATE MASS-FRAGMENT EMISSION IN HEAVY ION REACTIONS

I.M. Brâncuş*, J. Oehlschläger and J. Wentz

*Permanent address: Central Institute of Physics, IPNE
Bucharest (Romania)

Kernforschungszentrum Karlsruhe GmbH, Karlsruhe

Als Manuskript vervielfältigt
Für diesen Bericht behalten wir uns alle Rechte vor

Kernforschungszentrum Karlsruhe GmbH
Postfach 3640, 7500 Karlsruhe 1

ISSN 0303-4003

ABSTRACT

IMFREM: A COMPUTER PROGRAM FOR ANALYSES OF INTERMEDIATE MASS FRAGMENT EMISSION IN HEAVY ION REACTIONS.

The computer program IMFREM is based on a recently worked out extended sum-rule model (I.M. Brâncuş, KfK 4453 (to be published 1988)) for light and intermediate mass-fragment emission in heavy ion reactions. The report briefly describes the use of this program, the necessary input for the calculations of the element distribution and partial cross sections and gives a Fortran listing. The program includes a fitting routine FITEX for fast parameter adjustments. The use is demonstrated by an application to a specific example.

ZUSAMMENFASSUNG

INFREM: EIN RECHENPROGRAMM ZUR ANALYSE DER INTERMEDIATE MASS FRAGMENT EMISSION IN SCHWER-IONEN-REAKTIONEN.

Das Rechenprogramm INFREM basiert auf einem vor kurzem ausgearbeiteten Modell einer erweiterten Summen-Regel (I.M. Brâncuş, KfK 4453 (in Vorbereitung 1988)) für die Emission von leichten Teilchen und Fragmenten mittlerer Masse in Schwer-Ionen-Reaktionen. Die Benutzung des Programms, die notwendige Eingabe für die Berechnungen der Elementverteilungen und partiellen Wirkungsquerschnitte wird ausführlich beschrieben. Die Fortran-Liste ist vollständig mit angegeben. Das Programm enthält die Fit-Routine FITEX zum Anpassen von Parametern. Die Anwendung wird mit einem Beispiel verdeutlicht.

CONTENTS

1. INTRODUCTION AND BASIC FORMULAS

Assuming that complete and incomplete fusion, break-up and intermediate mass fragment (IMF) emission are competing reaction mechanisms and do proceed via a partial statistical equilibration, the following sum-rule /1/ is used for calculating the reaction cross sections for a particular exit reaction channel (i) :

$$N_l \left\{ \sum_{i=1}^{n} T_{l(i)} P(i) + \sum_{i=2}^{n} T_l' P(i) \right\} = 1 \qquad (1)$$

Here N_l are overall normalisation factors. The reaction cross section for each projectile product (i) is the sum of two contributions

$$\sigma^{tot}(i) = \sigma(i) + \sigma'(i) \qquad (2)$$

The term σ(i), representing the fusion (i = 1) and the incomplete fusion processes (i = 2, . . . , n), is given by the expression provided by the original sum-rule model /2/

$$\sigma(i) = \pi\lambda^2 \sum_{l=0}^{l_{max}} (2l+1) \frac{T_{l(i)} P(i)}{\sum_{i=1}^{n} T_{l(i)} P(i) + \sum_{i=2}^{n} T_l' P(i)} \qquad (3)$$

The second term σ' (i) describes the emission of light particles from the dinuclear system on its route to complete fusion :

$$\sigma'(i) = \pi\lambda^2 \sum_{l=0}^{l_{max}} (2l+1) \frac{T_l' P(i)}{\sum_{i=1}^{n} T_{l(i)} P(i) + \sum_{i=2}^{n'} T_l' P(i)} \qquad (4)$$

Thus, two forms for the transmission coefficient are used : *one* depending on the angular momenta in the entrance channel

$$T_{l(i)} = \left\{ 1 + exp \left| (l - l_{lim}(i)) / \Delta l \right| \right\}^{-1} \qquad (5)$$

and other depending on the critical value for fusion calculated as a dissipative process, identical for all the exit channels

$$T_l' = \left\{ 1 + exp \left| (l - l_{cr}^{dyn}) / \Delta l \right| \right\}^{-1} \qquad (6)$$

P (i) represents the reaction probability for each reaction channel

$$P(i) \propto exp \{ | Q_{gg}(i) - Q_c(i) | / T \} \qquad (7)$$

withT being the effective temperature, Q_{gg} the Q-value of the ground state, and Q_c the change of Coulomb interaction,

$$Q_c = (Z_1^f Z_2^f - Z_1^i Z_2^i)\, e^2 / R_c \tag{8}$$

R_c is the relative distance, where the transfer of charge takes place

$$R_c = R_{oc} \cdot (A_1^{1/3} + A_2^{1/3}) \quad , \tag{9}$$

and $\ell_{\ell im}$ is related to the critical angular momentum value by

$$l_{lim}(A_1 \; vs \; A_2) = \frac{A_1 A_2}{mA_1 + nA_2}\, l_{cr}(n \; vs \; A_2 \; or \; m \; vs \; A_1) \tag{10}$$

where

$$\left(l_{cr} + \frac{1}{2}\right)^2 = \mu \frac{(C_1 + C_2)^3}{\hbar^2} \left[4\pi\gamma \frac{C_1 C_2}{C_1 + C_2} - \frac{Z_1 Z_2 e^2}{(C_1 + C_2)^2} \right] \tag{11}$$

C_1, C_2 being the half-density radii,

$$C_i = R_i \left[1 - (b/R_i)^2 \pm \ldots \right], b = 1 \, fm, \tag{12}$$

$$R_i = 1.28 \, A_i^{1/3} - 0.76 + 0.8 \, A_i^{-1/3} \quad . \tag{13}$$

The calculations involve three parameters T, R_{0C} and $\Delta\ell$ which can be phenomenologically determined by the best fit to experimental data.

The model is described in detail in ref. 1. The following report gives the program for the calculations of the cross sections and introduces in its technical use. For conveniences of the user we add a worked out example.

2. MAIN PROGRAM IMFREM

2.1 SHORT SUMMARY OF THE STRUCTURE OF IMFREM

The computer program IMFREM for analyses of intermediate mass fragment emission is written in Fortran-77 and uses single precision variables. Actually it is running on a Siemens 7890 and an IBM 3090 computer, and in a more interactive way on personal computers compatible to IBM PC AT set ups.
After the Fortran-77 definitions of arrays and common blocks the program IMFREM needs the input line named "zero":

calculation character string, projectile-target string
(always 2 character strings)

The first character string defines the kind and the number of calculations, with or without parameter fit, only the fit procedure or or just a cross section calculation, and additionally which type of model is chosen. With the character string of the required calculation an integer variable ICALC is determined for a characterisation with which the calculations can be controlled easierly. ICALC can have only fifteen different and valid numbers. For each of these numbers the program requires now the next input line named "one":

masses & charges of projectile and target, and the
incident energy (always 5 real values)

After that it needs the next input line "two":

maximum value of the angular momentum confining the interval
of the sum in the l-space etc. (2 or 5 integer values)

After that it needs the next input line "three":

starting values for fit parameters or fixed values
(3 or 4 real values). But only the first two quantities
(i.e. TEMP, ROC) are fitted.

Now some physical and mathematical constants are defined. Then the input tables must be read from several files. Leading the Q_{gg}*-values for*

either the calculation of partial cross sections fixed
implemented as READ(2,*)
or the total cross sections fixed implemented as READ(3,*)

and after that the experimental data fixed implemented as Fortran statements with READ(4,). The errors of the experimental data points or their weights for the fit procedure are normalized in respect to the maximum. Both the files of Q_{gg}-values and the experimental data files contain a line which is suggested to use for comments, second the actual number of data points, and after that beginning with the third line in each file the data points:*

first: comment line
second: number of points
after that: data points

Then the input values and the experimental data points are printed out. After the successful call of FITEX some calculated tables are printed out. If the user requires only fitting of parameters the program stops here. If not, one of the following six types of cross section evaluations is performed: Either the original model, or the extended model or the extended modified model for the total cross sections or the corresponding calculation for partial cross sections. After the print-out of the tables the program stops.

2.2 DESCRIPTION OF ALL VARIABLES

ABS = Fortran-77 intrinsic function
AI = (int(A12-Z12) - Z12) / A12
ALAM = sqrt (ALAM2)
ALAM2 = 0.5 / (CL * ARED * ECM)
ALCR(i),i=1,IDB) = array for l_{cr}-values.
ALIM(i),i=1,IDB) = array for l_{lim}-values.
ALL = COMMON block containing the arrays Z3P, A3P, QGG, P, ALCR, ALIM, and PMOD
AMU = constant 931.478
ARED = reduced mass A1*A2/(A1+A2)
AREDI = subsidiary variable for reduced mass calculations
AT = subsidiary variable for differences of masses
ATAN = Fortran-77 intrinsic function
A1,A2 = mass of the projectile and the target
A12 = A1 + A2
A3 = subsidiary variable for A3P elements

A3P(i),i=1,IDB = masses for QGG-values
A4 = subsidiary variable for differences of masses
CHCALC*6 = character string with the calculation option
CL = $931.478 / 197.32^2$
CMODEL(i)*4,i=1,10 = strings for model characterisation
CPROTA*4 = character string with projectile-target combination
CT = either RT-1/RT or $ROT*AT^{1/3}$
C0 = subsidiary variable in the calculation of ALCR(i)
C1 = subsidiary variable for the expression R1 - 1 / R1
C2 = subsidiary variable for the expression R2 - 1 / R2
DD = additional term for the modified R_C
DEL = $\Delta 1$
DELD = $1 / \Delta 1$
ECM = ELAB * A2 / (A1 + A2), center of mass energy
ELAB = incident energy
EXP = Fortran-77 intrinsic function
E1 = constant 1.44
FVAL(i),i=1,MPTX = actual function values fitted with FITEX
GAM = $0.95 * (1 - 1.78 * AI^2)$
HC,HC2 = constant 197.32 and constant 197.32^2
I = loop variable, subsidiary variable
IABS = Fortran-77 intrinsic function
IC = subsidiary variable during the check of CHCALC
ICALC = integer value corresponding to CHCALC
IDA = dimension 20
IDB = dimension 200
II,IO,IP = loop variable, subsidiary variable
IPIN = maximum number of reaction channels of interest for the second term
IPMAX = maximum number of reaction channels
ISW(i,j),i=1,2;j=0,IDA = integer array with the first number of terms of isotopes (i.e. ISW(1,*)) and the number of terms to be summed (i.e. ISW(2,*)).
ISWAG(i,j),i=1,2;j=0,IDA = with numbers of terms to be summed; for combination "LIAG", to get

element (1) sum nr. (3),(4),(5),
element (2) sum nr. (6),(7),(8),
element (3) sum nr. (9),
element (4) sum nr. (11),
element (5) sum nr. (12),(13),(14),(15),

```
            element  (6) sum nr. (16),(17),(18),(19),(20),
            element  (7) sum nr. (21),(22),(23),(24),
            element  (8) sum nr. (25),(26),(27),(28),
            element  (9) sum nr. (29),(30),(31),(32),(33),
            element (10) sum nr. (34),(35),(36),(37),(38),
            element (11) sum nr. (39),(40),(41),(42),
            element (12) sum nr. (43),(44),(45),(46),(47),
            element (13) sum nr. (48),(49),(50),(51),(52),
            element (14) sum nr. (53),(54),(55),(56),(57),(58),
            element (15) sum nr. (59),(60),(61),(62),(63),(64).
ISWCU(i,j),i=1,2;j=0,IDA = with numbers of terms to be summed;
        for combination "LICU", to get
            element  (4) sum nr. (11),
            element  (5) sum nr. (12),(13),(14),
            element  (6) sum nr. (15),(16),(17),
            element  (7) sum nr. (18),(19),(20),(21),
            element  (8) sum nr. (22),(23),(24),(25),
            element  (9) sum nr. (26),(27),(28),(29),
            element (10) sum nr. (30),(31),(32),
            element (11) sum nr. (33),(34),(35),(36),
            element (12) sum nr. (37),(38),(39),(40),(41),
            element (13) sum nr. (42),(43),(44),(45),
            element (14) sum nr. (56),(47),(48),(49),(50),(51),
            element (15) sum nr. (52),(53),(54),(55),(56),(57).
ISWTI(i,j),i=1,2;j=0,IDA = with numbers of terms to be summed;
        for combination "LITI", to get
            element  (1) sum nr. (3),(4),
            element  (2) sum nr. (6),(7)
            element  (3) sum nr. (9),
            element  (4) sum nr. (10),(12)
            element  (5) sum nr. (13),(14),(15),(16),
            element  (6) sum nr. (17),(18),(19),(20),
            element  (7) sum nr. (21),(22),(23),(24),
            element  (8) sum nr. (25),(26),(27),(28),
            element  (9) sum nr. (29),(30),(31),(32),
            element (10) sum nr. (33),(34),(35),
            element (11) sum nr. (36),(47),(48).
IWORK(i),i=1,IWX = integer working array of FITEX
IWX     = MPAX + 5
```

L = loop variable, subsidiary variable
LCRF = value of the critical angular momentum for fusion with dissipation, l_{cr}^{dyn} (calculated with a dynamical model)
LFIN = maximum value of l_{max} for the second term
LL = loop variable, subsidiary variable
LMAX = maximum value of l_{max}, the angular momentum confining the interval of the sum in the l-space
LOG = Fortran-77 intrinsic function
MIT = maximum number of iterations of FITEX
MOD = Fortran-77 intrinsic function
MPA = actual number of parameters to be fitted
MPAX = maximum number of fit parameters of FITEX
MPB = subsidiary variable to print more results of the fit
MPR = subsidiary variable to print more results of the fit
MPT = number of experimental data points
MPTX = maximum number of experimental data points of FITEX
MRET = return code, i.e. convergence criterum of FITEX
MWP = subsidiary variable to print more results of the fit
MWX = upper limit of the lenght of array WORK
P(i),i=1,IDB = array for P-values
PACC(i),i=1,MPAX = accuracies of fit parameters
PAR = COMMON block with TEMP, ROC, DD, DEL
PARM(i),i=1,MPAX = actual values of the fit parameters
PERR(i),i=1,MPAX = calculated errors of the values of the fitted parameters
PI = constant π
PILA = 10 • PILAM
PILAM = π • ALAM2
PI4 = 4 π
PMOD(i),i=1,IDB = array for modified P-values
PM1 = -1/3
PVAR(i,j),i=1,MPAX;j=1,MPAX = subsidiary array to calculate the errors of the fit parameters
P1 = 1/3
QC = change of Coulomb interaction energy
QCMOD = value QC of the modified calculations
QGG(i),i=1,IDB = Q-values of the ground state
RC = effective relative distance R_C
RCMOD = value RC of the modified calculations
RT = subsidiary variable

ROC = radius R_{oc}
ROT = constant 0.764
R1,R2 = subsidiary variable
SEC(i),i=1,IDA = experimental data values
SECER(i),i=1,IDA = 20 % error of the experimental data
SECFI(i),i=1,IDA = actual theoretical data during the fit procedure with FITEX, calculated in FITFUN
SECOR(i),i=1,IDA = actual values of array WILORG
SECRE(i),i=1,IDA = actual values of array WILRES
SECT = COMMON block with array SIG, SIGRES, and SIGTOT
SECTO(i),i=1,IDA = actual values of array WILTOT
SIG(i),i=1,IDB = reaction cross sections original parts, normalized linewise summation of array SIGL
SIGL(i,j),i=1,IDB,j=1,IDB = partial cross sections original part
SIGNOR(i,j),i=1,IDB,j=1,IDB = new partial cross sections
SIGRES(i),i=1,IDB = reaction cross sections new parts
SIGTOT(i),i=1,IDB = total reaction cross sections
SM = summation of reaction cross sections original part
SMRES = summation of reaction cross sections new part
SQRT = Fortran-77 intrinsic function
SUM(i),i=1,IDB = columnwise summation of array SIGL
TEMP,TEMPD = temperature, 1 / TEMP
TL(i),i=1,IDB = transmission coefficients
TLRES(i),i=1,IDB = transmission coefficients, modified form
VAL = COMMON block with special quantities
WILORG(i),i=1,IDA = sum of isotopes, cross section original part
WILRES(i),i=1,IDA = sum of isotopes, new parts of reaction cross sections
WILSUM(i),i=1,IDA = total reaction cross sections
WORK(i),i=1,MWX = real working array of FITEX
WSUB = subsidiary variable for WORK(3)
XSEC(i),i=1,IDA = abszissas of the experimental data
ZLP(i),i=1,IDB = real values 2•i+1
ZT = subsidiary variable for differences of charges
Z1,Z2 = charge of projectile and target
Z12 = Z1 + Z2
Z3 = subsidiary variable for Z3P elements
Z3P(i),i=1,IDB = charges for QGG-values
Z4 = subsidiary variable for differences of charges

2.3 LISTING OF THE MAIN PROGRAM IMFREM

```
C-------------------------------------------------------------
C AVAILABLE CALCULATION OPTIONS                        INTERNAL
C *****************************                           CODE
C                                                        NUMBER
C FI - ORIGINAL MODEL, FITEX-FIT                           100
C F1 - EXTENDED MODEL, FITEX-FIT                            10
C F2 - EXTENDED MODEL, MODIFIED, FITEX-FIT                  20
C WI - ORIGINAL MODEL, TOTAL REACTION CROSS SECTION          3
C W1 - EXTENDED MODEL, TOTAL REACTION CROSS SECTION          1
C W2 - EXTENDED MODEL, MODIFIED, TOTAL REACTION CR.SE.       2
C SI - ORIGINAL MODEL, PARTIAL CROSS SECTION                 8
C S1 - EXTENDED MODEL, PARTIAL CROSS SECTION                 6
C S2 - EXTENDED MODEL, MODIFIED, PARTIAL CROSS SECTION       7
C-------------------------------------------------------------
C VALID CALCULATIONS                                INTERNAL
C *****************                                   CODE
C                                                    NUMBER
C      'W1'       FOR  W1  ONLY                         1
C      'W2'       FOR  W2  ONLY                         2
C      'WI'       FOR  WI  ONLY                         3
C      'S1'       FOR  S1  ONLY                         6
C      'S2'       FOR  S2  ONLY                         7
C      'SI'       FOR  SI  ONLY                         8
C      'FI'       FOR  FI  ONLY                       100
C      'FIWI'     FOR  FI  AND  WI                    103
C      'FISI'     FOR  FI  AND  SI                    108
C      'FIF1'     FOR  (FI + F1)                      110
C      'FIF1W1'   FOR  (FI + F1)  AND  W1             111
C      'FIF1S1'   FOR  (FI + F1)  AND  S1             116
C      'FIF2'     FOR  (FI + F2)                      120
C      'FIF2W2'   FOR  (FI + F2)  AND  W2             122
C      'FIF2S2'   FOR  (FI + F2)  AND  S2             127
C-------------------------------------------------------------
      PROGRAM IMFREM
      PARAMETER (IDA=20,IDB=200)
      PARAMETER (MPTX=20,MPAX=5,IWX=MPAX+5,MWX=14+
     +   MPAX*(MPAX+5)/2+(MPTX+MPAX+1)*(MPAX+2)+MPTX*MPAX)
      CHARACTER CHCALC*6,CPROTA*4,CMODEL(10)*4
```

```
      DIMENSION PVAR(MPAX,MPAX),PERR(MPAX),PARM(MPAX)
      DIMENSION PACC(MPAX),WORK(MWX),FVAL(MPTX),IWORK(IWX)
      COMMON /ALL/Z3P(IDB),A3P(IDB),QGG(IDB)
     +   ,P(IDB),ALCR(IDB),ALIM(IDB),PMOD(IDB),ZLP(IDB)
      COMMON /SECT/SIG(IDB),SIGRES(IDB),SIGTOT(IDB)
     +   ,WILORG(IDA),WILRES(IDA),WILSUM(IDA)
      COMMON /SFIT/XSEC(IDA),SEC(IDA),SECFI(IDA)
     +   ,SECOR(IDA),SECRE(IDA),SECTO(IDA),SECER(IDA)
     +   ,SUM(IDB),SIGL(IDB,IDB),SIGNOR(IDB,IDB)
      COMMON /VAL/A1,Z1,A2,Z2,ELAB,ECM,A12,Z12,GAM,CL
     +   ,ALAM,PILAM,P1,PM1,ROT,RC,RCMOD,PI4,E1
     +   ,IPMAX,IPIN,LCRF,LFIN,LMAX,ICALC
      COMMON /PAR/TEMP,ROC,DEL,DD
      DATA CHCALC/'      '/,CPROTA/'    '/
      DATA CMODEL/'"W1"','"W2"','"WI"','    ','    ',
     +           '"S1"','"S2"','"SI"','    ','    '/
C-ZEROTH LINE - - - - - - - - - - - - - - - - - - - - - - - - - -
      READ(*,'(A80)')
      READ(*,*) CHCALC,CPROTA
      WRITE(*,'(/11X,23(''='')/11X,''START OF PROGRAM IMFREM''/
     +11X,54(''='')/11X,''VERSION '',A6,'' WITH PROJECTILE-'',
     +''TARGET COMBINATION '',A4/11X,54(''=''))') CHCALC,CPROTA
      ICALC=0
      IF (CHCALC(1:2).EQ.'FI') ICALC=ICALC+100
      IC=ICALC/50
      IF (CHCALC(IC+1:IC+2).EQ.'WI') ICALC=ICALC+3
      IF (CHCALC(IC+1:IC+2).EQ.'SI') ICALC=ICALC+8
      IF (CHCALC(IC+1:IC+2).EQ.'F1') ICALC=ICALC+10
      IF (CHCALC(IC+1:IC+2).EQ.'F2') ICALC=ICALC+20
      IF (ICALC.GE.110) IC=IC+2
      IF (CHCALC(IC+1:IC+2).EQ.'W1') ICALC=ICALC+1
      IF (CHCALC(IC+1:IC+2).EQ.'W2') ICALC=ICALC+2
      IF (CHCALC(IC+1:IC+2).EQ.'S1') ICALC=ICALC+6
      IF (CHCALC(IC+1:IC+2).EQ.'S2') ICALC=ICALC+7
      WRITE(*,'(11X,''I C A L C ='',I4)') ICALC
      IF (ICALC.LE.0.OR.ICALC.GT.127) THEN
         WRITE(*,'(11X,''E R R O R :  VERSION NUMBER'',I4,
     +       '' INVALID'')') ICALC
         GOTO 199
```

```
         ENDIF
C-FIRST LINE  - - - - - - - - - - - - - - - - - - - - - - - - -
      READ(*,'(A80)')
      READ(*,*) A1,Z1,A2,Z2,ELAB
C-SECOND LINE - - - - - - - - - - - - - - - - - - - - - - - - -
      READ(*,'(A80)')
      IF (ICALC.EQ.100.OR.MOD(ICALC,5).EQ.3) THEN
         READ(*,*) LMAX,IPMAX
      ELSE
         READ(*,*) LMAX,LFIN,IPMAX,IPIN,LCRF
      ENDIF
C-THIRD LINE  - - - - - - - - - - - - - - - - - - - - - - - - -
      READ(*,'(A80)')
      IF (ICALC.EQ.120.OR.MOD(ICALC,5).EQ.2) THEN
         READ(*,*) TEMP,ROC,DD,DEL
      ELSE
         READ(*,*) TEMP,ROC,DEL
      ENDIF
C - - - - CONSTANTS - - - - - - - - - - - - - - - - - - - - - - -
      ROT=0.764
      PI=4.*ATAN(1.)
      PI4=4.*PI
      P1=1./3.
      PM1=-P1
      A12=A1+A2
      Z12=Z1+Z2
      I=A12-Z12
      AI=(I-Z12)/A12
      GAM=.95*(1.-1.78*AI*AI)
      E1=1.44
      HC=197.32
      HC2=HC*HC
      AMU=931.478
      ARED=A1*A2/(A1+A2)
      CL=AMU/HC2
      ECM=ELAB*A2/(A1+A2)
      ALAM2=0.5/(CL*ARED*ECM)
      ALAM=SQRT(ALAM2)
      PILAM=PI*ALAM2
      PILA=10.*PILAM
```

```
      DELD=1./DEL
      TEMPD=1./TEMP
      DO  100  I=1,IDB
  100 ZLP(I)=1.+2.*I
      RC=ROC*(A1**P1+A2**P1)
      RCMOD=RC
      IF (ICALC.EQ.120.OR.MOD(ICALC,5).EQ.2)
     +   RCMOD=1.225*(A1**P1+A2**P1)+DD
C-READ QGG-DATA - - - - - - - - - - - - - - - - - - - - - - - - -
      IF (MOD(ICALC,10).GE.6) THEN
         READ(2,'(A80)')
         READ(2,*) IP
         DO  101  I=1,IP
  101    READ(2,*,END=103) Z3P(I),A3P(I),QGG(I)
      ELSE
         READ(3,'(A80)')
         READ(3,*) IP
         DO  102  I=1,IP
  102    READ(3,*,END=103) Z3P(I),A3P(I),QGG(I)
      ENDIF
C-READ EXPERIMENTAL DATA  - - - - - - - - - - - - - - - - - - - -
  103 READ(4,'(A80)')
      READ(4,*) MPT
      IO=1
      DO  104  I=1,MPT
         READ(4,*,END=105) XSEC(I),SEC(I)
         IF (I.GT.1.AND.SEC(I).GT.SEC(I-1)) IO=I
  104 CONTINUE
  105 CONTINUE
      DO  106  I=1,MPT
         SECER(I)=SEC(I)/SEC(IO)
  106 CONTINUE
C - - - - PRINT-OUT OF INPUT VALUES AND DATA  - - - - - - - - - -
      WRITE(*,'(/6X,''INPUT VALUES:''/)')
      WRITE(*,601) A1,Z1,A2,Z2,ELAB
      IF (ICALC.EQ.100.OR.MOD(ICALC,5).EQ.3) THEN
         WRITE(*,602) LMAX,IPMAX
      ELSE
         WRITE(*,603) LMAX,LFIN,IPMAX,IPIN,LCRF
      ENDIF
```

```
      IF (ICALC.EQ.120.OR.MOD(ICALC,5).EQ.2) THEN
         WRITE(*,604) TEMP,ROC,DD,DEL
      ELSE
         WRITE(*,605) TEMP,ROC,DEL
      ENDIF
      WRITE(*,606) (I,XSEC(I),SEC(I),SECER(I),I=1,MPT)
      WRITE(*,'(/6X,''ADDITIONAL CONSTANTS:''/)')
      WRITE(*,607) GAM,ARED,ALAM,ECM
      IF (ICALC.EQ.120.OR.MOD(ICALC,5).EQ.2) THEN
         WRITE(*,608) RC,RCMOD,ROT
      ELSE
         WRITE(*,609) RC,ROT
      ENDIF
      IF (ICALC.LT.100) GOTO 118
C - - - - - FIT PROCEDURE WITH FITEX (TEMP, ROC) - - - - - - - - -
      DO  110  I=1,MPAX
         PACC(I)=0.002
  110 CONTINUE
      DO  111  I=1,MWX
         WORK(I)=-9.9999
  111 CONTINUE
      DO  112  I=1,IWX
         IWORK(I)=0
  112 CONTINUE
      PARM(1)=TEMP
      IF (ICALC.EQ.120.OR.MOD(ICALC,5).EQ.2) THEN
         PARM(2)=DD
      ELSE
         PARM(2)=ROC
      ENDIF
      MPA=2
      MPR=1
      MWP=1
      MRET=0
      MIT=100
      MPB=IABS(MPR)
      WORK(1)=1./PACC(1)
      WORK(2)=0.
      WORK(3)=0.
      IWORK(1)=MPA+1
```

```
      IWORK(2)=MIT
      IWORK(3)=1
C - - - - - - IMPLICIT LOOP WITH CALL OF FITEX - - - - - - - - -
  113 CONTINUE
      TEMP=PARM(1)
      IF (ICALC.EQ.120.OR.MOD(ICALC,5).EQ.2) THEN
         DD=PARM(2)
         RCMOD=1.225*(A1**P1+A2**P1)+DD
      ELSE
         ROC=PARM(2)
      ENDIF
      CALL FITFUN(MPT,MPA)
      WORK(4)=0.
      DO  114  I=1,MPT
         FVAL(I)=LOG(100.*SEC(I))-LOG(100.*SECFI(I))
         WORK(4)=WORK(4)+FVAL(I)*FVAL(I)
  114 CONTINUE
      IF (.NOT.(IWORK(3).NE.MWP.AND.IWORK(3).NE.1.AND.
     +   MPB.NE.1)) THEN
         WRITE(*,610) IWORK(3),WORK(4),IWORK(4),WORK(3)
         WRITE(*,611) (PARM(I),I=1,MPA)
         MWP=MWP+MPB
         IF (MPR.GT.0) WRITE(*,612) (FVAL(I),I=1,MPT)
         ENDIF
      CALL FITEX(MRET,MPT,MPA,FVAL,PARM,PACC,WORK,IWORK)
      IF (MRET.EQ.1) GOTO 113
C - - - - - - END OF FITEX USE; PRINT-OUT OF RESULTS - - - - - -
      TEMP=PARM(1)
      TEMPD=1./TEMP
      IF (ICALC.EQ.120.OR.MOD(ICALC,5).EQ.2) THEN
         DD=PARM(2)
         RCMOD=1.225*(A1**P1+A2**P1)+DD
      ELSE
         ROC=PARM(2)
      ENDIF
      RC=ROC*(A1**P1+A2**P1)
      WRITE(*,610) IWORK(3),WORK(4),IWORK(4),WORK(3)
      WRITE(*,611) (PARM(I),I=1,MPA)
      IF (MPR.GT.0) THEN
         WRITE(*,612) (FVAL(I),I=1,MPT)
```

```
         IF (MRET*(MRET-3).EQ.0.AND.WORK(5).GT.0.) THEN
         WRITE(*,'(11X,''STANDARD ERRORS'')')
         WRITE(*,'(11X,1P,5E13.5)') (WORK(4+I),I=1,MPA)
         WRITE(*,'(11X,''ERROR ENHANCEMENTS'')')
         WRITE(*,'(11X,1P,5E13.5)') (WORK(4+MPA+I),I=1,MPA)
         WRITE(*,'(11X,''ERROR CORRELATION MATRIX'')')
         L=4+MPA+MPA
         DO  115  IO=1,MPA
            II=L+1
            L=L+IO
            WRITE(*,'(11X,1P,5E13.5)') (WORK(I),I=II,L)
  115    CONTINUE
         ENDIF
         ENDIF
      WRITE(*,613) MRET
      IF (IABS(MIT).GT.1) THEN
      L=4+MPA+MPA
      WSUB=1.
      IF (WORK(3).NE.0.) WSUB=WSUB/WORK(3)
      DO  116  I=1,MPA
         II=L+1
         L=L+I
         IP=0
         DO  116  IO=II,L
            IP=IP+1
            PVAR(IP,I)=WORK(IO)*WORK(4+I)*WORK(4+IP)*WSUB*WSUB
            PVAR(I,IP)=PVAR(IP,I)
  116    CONTINUE
         DO  117  I=1,MPA
            PVAR(I,I)=(WORK(4+I)*WSUB)**2
            PERR(I)=SQRT(PVAR(I,I))
  117    CONTINUE
         WRITE(*,614) (I,PARM(I),PERR(I),I=1,MPA)
      ENDIF
C - - - - - PRINT-OUT AFTER FIT PROCEDURE - - - - - - - - - - -
      IF (MOD(ICALC,10).EQ.0) THEN
         IF (ICALC.EQ.120.OR.MOD(ICALC,5).EQ.2) THEN
            WRITE(*,615)(IP,Z3P(IP),A3P(IP),QGG(IP),P(IP)
     +           ,PMOD(IP),ALCR(IP),ALIM(IP),IP=1,IPMAX)
         ELSE
```

```
         WRITE(*,616)(IP,Z3P(IP),A3P(IP),QGG(IP),P(IP)
     +          ,ALCR(IP),ALIM(IP),IP=1,IPMAX)
      ENDIF
      IF (ICALC.EQ.100.OR.MOD(ICALC,5).EQ.3) THEN
         WRITE(*,617)(IP,Z3P(IP),A3P(IP),SIG(IP),IP=1,IPMAX)
      ELSE
         WRITE(*,618)(IP,Z3P(IP),A3P(IP),SIG(IP),SIGRES(IP)
     +          ,SIGTOT(IP),IP=1,IPMAX)
      ENDIF
      ENDIF
C - - - - - CALCULATE TOTAL OR PARTIAL CROSS SECTIONS - - - - - -
  118 CONTINUE
      ICALC=MOD(ICALC,10)
      IF (ICALC.EQ.0) GOTO 199
      WRITE(*,'(/11X,A4)') CMODEL(ICALC)
      CALL FITFUN(MPT,MPA)
C - - - - - - - "W1"
      IF (MOD(ICALC,10).EQ.1) THEN
         WRITE(*,616)(I,Z3P(I),A3P(I),QGG(I),P(I)
     +    ,ALCR(I),ALIM(I),I=1,IPMAX)
         WRITE(*,618)(I,Z3P(I),A3P(I),SIG(I),SIGRES(I)
     +    ,SIGTOT(I),I=1,IPMAX)
         WRITE(*,620) (I,XSEC(I),SEC(I),SECOR(I),SECRE(I)
     +    ,SECTO(I),I=1,MPT)
         ENDIF
C - - - - - - - "W2"
      IF (MOD(ICALC,10).EQ.2) THEN
         WRITE(*,615) (I,Z3P(I),A3P(I),QGG(I),P(I),PMOD(I)
     +    ,ALCR(I),ALIM(I),I=1,IPMAX)
         WRITE(*,618) (I,Z3P(I),A3P(I),SIG(I),SIGRES(I)
     +    ,SIGTOT(I),I=1,IPMAX)
         WRITE(*,620) (I,XSEC(I),SEC(I),SECOR(I),SECRE(I)
     +    ,SECTO(I),I=1,MPT)
         ENDIF
C - - - - - - - "WI"
      IF (MOD(ICALC,10).EQ.3) THEN
         WRITE(*,616) (I,Z3P(I),A3P(I),QGG(I),P(I)
     +    ,ALCR(I),ALIM(I),I=1,IPMAX)
         WRITE(*,617) (I,Z3P(I),A3P(I),SIG(I),I=1,IPMAX)
         WRITE(*,619) (I,XSEC(I),SEC(I),SECOR(I),I=1,MPT)
```

```
      ENDIF
C - - - - - - - "S1"
    IF (MOD(ICALC,10).EQ.6) THEN
      WRITE(*,616) (I,Z3P(I),A3P(I),QGG(I),P(I)
     +  ,ALCR(I),ALIM(I),I=1,IPMAX)
      WRITE(*,621)
      WRITE(*,622)
      WRITE(*,626) (LL,SIGL(LL,1),SIGL(LL,4),SIGL(LL,5)
     +  ,SIGL(LL,6),SIGL(LL,7),SIGL(LL,8),SIGL(LL,9)
     +  ,LL=1,LMAX)
      WRITE(*,623)
      WRITE(*,626) (LL,SIGL(LL,10),SIGL(LL,11),SIGL(LL,12)
     +  ,SIGL(LL,13),SIGL(LL,14),SIGL(LL,15),SIGL(LL,16)
     +  ,LL=1,LMAX)
      WRITE(*,624)
      WRITE(*,626) (LL,SUM(LL),SIGNOR(LL,4),SIGNOR(LL,5)
     +  ,SIGNOR(LL,6),SIGNOR(LL,7),SIGNOR(LL,8),SIGNOR(LL,9)
     +  ,LL=1,LMAX)
      WRITE(*,625)
      WRITE(*,626) (LL,SIGNOR(LL,10),SIGNOR(LL,11)
     +   ,SIGNOR(LL,12),SIGNOR(LL,13),SIGNOR(LL,14)
     +   ,SIGNOR(LL,15),SIGNOR(LL,16),LL=1,LMAX)
      WRITE(*,618) (I,Z3P(I),A3P(I),SIG(I),SIGRES(I)
     +  ,SIGTOT(I),I=1,IPMAX)
      WRITE(*,620) (I,XSEC(I),SEC(I),SECOR(I),SECRE(I)
     +  ,SECTO(I),I=1,MPT)
      ENDIF
C - - - - - - - "S2"
    IF (MOD(ICALC,10).EQ.7) THEN
      WRITE(*,615) (I,Z3P(I),A3P(I),QGG(I),P(I),PMOD(I)
     +  ,ALCR(I),ALIM(I),I=1,IPMAX)
      WRITE(*,621)
      WRITE(*,622)
      WRITE(*,626) (LL,SIGL(LL,1),SIGL(LL,4),SIGL(LL,5)
     +  ,SIGL(LL,6),SIGL(LL,7),SIGL(LL,8),SIGL(LL,9)
     +  ,LL=1,LMAX)
      WRITE(*,623)
      WRITE(*,626) (LL,SIGL(LL,10),SIGL(LL,11),SIGL(LL,12)
     +  ,SIGL(LL,13),SIGL(LL,14),SIGL(LL,15),SIGL(LL,16)
     +  ,LL=1,LMAX)
```

```
      WRITE(*,624)
      WRITE(*,626) (LL,SUM(LL),SIGNOR(LL,4),SIGNOR(LL,5)
     +  ,SIGNOR(LL,6),SIGNOR(LL,7),SIGNOR(LL,8),SIGNOR(LL,9)
     +  ,LL=1,LMAX)
      WRITE(*,625)
      WRITE(*,626) (LL,SIGNOR(LL,10),SIGNOR(LL,11)
     +   ,SIGNOR(LL,12),SIGNOR(LL,13),SIGNOR(LL,14)
     +   ,SIGNOR(LL,15),SIGNOR(LL,16),LL=1,LMAX)
      WRITE(*,618) (I,Z3P(I),A3P(I),SIG(I),SIGRES(I)
     +  ,SIGTOT(I),I=1,IPMAX)
      WRITE(*,620) (I,XSEC(I),SEC(I),SECOR(I),SECRE(I)
     +  ,SECTO(I),I=1,MPT)
      ENDIF
C - - - - - - - "SI"
   IF (MOD(ICALC,10).EQ.8) THEN
      WRITE(*,616) (I,Z3P(I),A3P(I),QGG(I),P(I)
     +  ,ALCR(I),ALIM(I),I=1,IPMAX)
      WRITE(*,621)
      WRITE(*,622)
      WRITE(*,626) (LL,SIGL(LL,1),SIGL(LL,4),SIGL(LL,5)
     +  ,SIGL(LL,6),SIGL(LL,7),SIGL(LL,8),SIGL(LL,9)
     +  ,LL=1,LMAX)
      WRITE(*,623)
      WRITE(*,626) (LL,SIGL(LL,10),SIGL(LL,11),SIGL(LL,12)
     +  ,SIGL(LL,13),SIGL(LL,14),SIGL(LL,15),1./SUM(LL)
     +  ,LL=1,LMAX)
      WRITE(*,617) (I,Z3P(I),A3P(I),SIG(I),I=1,IPMAX)
      WRITE(*,619) (I,XSEC(I),SEC(I),SECOR(I),I=1,MPT)
      ENDIF
C - - - - END OF PROGRAM IMFREM - - - - - - - - - - - - - - -
  199 WRITE(*,'(/11X,54(''='')/11X,''VERSION '',A6,'' WITH '',
     +   ''PROJECTILE-TARGET COMBINATION '',A4/11X,54(''='')/
     +   11X,''OF PROGRAM IMFREM ENDED''/11X,23(''='')/)')
     +   CHCALC,CPROTA
      STOP
C - - - - FORMAT SPECIFICATIONS
  601 FORMAT(6X,'A1=',1P,E11.4,' Z1=',E11.4,' A2=',E11.4
     + ,' Z2=',E11.4/6X,'ELAB=',E11.4)
  602 FORMAT(6X,'LMAX=',I4,1X,'IPMAX=',I4)
  603 FORMAT(6X,'LMAX=',I4,1X,'LFIN=',I4,1X,'IPMAX=',I4
```

```
   + ,1X,'IPIN=',I4,1X,'LCRF=',I4)
604 FORMAT(6X,'TEMP=',F6.3,1X,'ROC=',F6.3,1X,'ROT=',F6.3,1X
   + ,'DEL=',F6.3)
605 FORMAT(6X,'TEMP=',F6.3,1X,'ROC=',F6.3,1X,'DEL=',F6.3)
606 FORMAT(/6X,'EXPERIMENTAL DATA:'//11X,'NUMBER',5X,'CANAL'
   +,10X,'SECTION',10X,'ERROR'/250(7X,0P,I8,1P,3E16.7,:,/))
607 FORMAT(6X,'GAM=',1P,E12.5,' ARED=',E12.5,' LAM=',E12.5
   + ,' ECM=',E12.5)
608 FORMAT(6X,'RC=',E12.5,1X,'RCMOD=',E12.5,1X,'ROT=',F6.3)
609 FORMAT(6X,'RC=',E12.5,1X,'ROT=',F6.3)
610 FORMAT(/0P,I8,' ITERATION   W(4) =',1P,E11.4,
   +    '  IW(4) =',0P,I4,'  W(3) =',1P,E11.4)
611 FORMAT(11X,'FIT PARAMETERS'/250(11X,1P,5E16.5,:,/))
612 FORMAT(11X,'FUNCTIONS'/250(11X,1P,5E13.5,:,/))
613 FORMAT(/11X,'CONVERGENCE WITH RETURN CODE',I2)
614 FORMAT(/11X,'FINAL PARAMETERS AT THE END OF SEARCH'/13X,
   +'NUMBER',7X,'VALUE',10X,'ERROR',5(/11X,0P,I6,1P,2E16.5))
615 FORMAT(/3X,'CAN',4X,'Z3',4X,'A3',5X,'QGG',11X,'P',9X,
   +'PMOD',8X,'LCR',9X,'LIM'/250(0P,I6,2F6.1,1P,5E12.4,:,/))
616 FORMAT(/3X,'CAN',5X,'Z3',6X,'A3',8X,'QGG',12X,'P'
   + ,12X,'LCR',11X,'LIM'/250(0P,I6,2F8.2,1P,4E14.5,:,/))
617 FORMAT(/10X,'NR.CANAL',5X,'Z',13X,'A',10X,'SIG(MB)',
   + 250(/12X,0P,I3,1P,3E14.5))
618 FORMAT(/3X,'NR.CANAL',4X,'Z',12X,'A',9X,'SIG(MB)',7X
   + ,'SIGRES',7X,'SIGTOT',250(/0P,I8,1P,5E13.5))
619 FORMAT(/11X,'CANAL',6X,'Z',11X,'SIGEXP',8X,'SIG(MB)',
   + 250(/12X,0P,I3,1P,3E14.5))
620 FORMAT(/4X,'NR.FIT',5X,'Z',9X,'SIG(MB)',7X,'SIGIN',7X
   + ,'SIGRES',7X,'SIGTOT',250(/0P,I8,1P,5E13.5))
621 FORMAT(/1X,'PARTIAL CROSS-SECTIONS')
622 FORMAT(/2X,'L',4X,'SIGLF',6X,'SIGL2',6X,'SIGL3',6X
   + ,'SIGL4',6X,'SIGL5',6X,'SIGL6',6X,'SIGL7')
623 FORMAT(/2X,'L',4X,'SIGL8',6X,'SIGL9',6X,'SIGL10',5X
   +,'SIGL11',5X,'SIGL12',5X,'SIGL13',5X,'NOR')
624 FORMAT(/2X,'L',5X,'SUM',7X,'SNOR2',6X,'SNOR3',6X
   + ,'SNOR4',6X,'SNOR5',6X,'SNOR6',6X,'SNOR7')
625 FORMAT(/2X,'L',4X,'SNOR8',6X,'SNOR9',6X,'SNOR10',5X
   + ,'SNOR11',5X,'SNOR12',5X,'SNOR13',5X,'SNOR14')
626 FORMAT(I3,1P,7E11.3)
    END
```

2.4. LISTING OF THE SUBROUTINE TO EVALUATE THE FUNCTION

```
C---------------------------------------------------------------
C      VALUES OF THE DIFFERENT SUM RULE CALCULATIONS
C---------------------------------------------------------------
      SUBROUTINE FITFUN(MPT,MPA)
      PARAMETER (IDA=20,IDB=200)
      DIMENSION TL(IDB),TLRES(IDB),ISW(2,0:IDA)
      DIMENSION ISWAG(2,0:IDA),ISWCU(2,0:IDA),ISWTI(2,0:IDA)
      COMMON /ALL/Z3P(IDB),A3P(IDB),QGG(IDB)
     +   ,P(IDB),ALCR(IDB),ALIM(IDB),PMOD(IDB),ZLP(IDB)
      COMMON /SECT/SIG(IDB),SIGRES(IDB),SIGTOT(IDB)
     +   ,WILORG(IDA),WILRES(IDA),WILSUM(IDA)
      COMMON /SFIT/XSEC(IDA),SEC(IDA),SECFI(IDA)
     +   ,SECOR(IDA),SECRE(IDA),SECTO(IDA),SECER(IDA)
     +   ,SUM(IDB),SIGL(IDB,IDB),SIGNOR(IDB,IDB)
      COMMON /VAL/A1,Z1,A2,Z2,ELAB,ECM,A12,Z12,GAM,CL
     +   ,ALAM,PILAM,P1,PM1,ROT,RC,RCMOD,PI4,E1
     +   ,IPMAX,IPIN,LCRF,LFIN,LMAX,ICALC
      COMMON /PAR/TEMP,ROC,DEL,DD
      DATA ISW/42*0/
C-ONLY FOR "LIAG"
      DATA ISWAG/ 1,15, 3,3, 6,3, 9,0, 11,1, 12,4, 16,5, 21,4,
     +   25,4, 29,5, 34,5, 39,4, 43,5, 48,5, 53,6, 59,6, 10*0/
C-ONLY FOR "LICU"
      DATA ISWCU/ 4,15, 3,0, 6,0, 9,0, 11,1, 12,3, 15,3, 18,4,
     +   22,4, 26,4, 30,3, 33,4, 37,5, 42,4, 46,6, 52,6, 10*0/
C-ONLY FOR "LITI"
      DATA ISWTI/ 1,11, 3,2, 6,2, 9,0, 10,3, 13,4, 17,4, 21,4,
     +   25,4, 29,4, 33,3, 36,3, 18*0/
C COPY TO GENERAL ARRAY ISW FOR LATER USE
      IF (Z2.EQ.22.) THEN
         ISW(1,0)=ISWTI(1,0)
         ISW(2,0)=ISWTI(2,0)
         DO  122  IO=ISW(1,0),ISW(2,0)
            ISW(1,IO)=ISWTI(1,IO)
            ISW(2,IO)=ISWTI(2,IO)
  122    CONTINUE
         ENDIF
      IF (Z2.EQ.29.) THEN
```

```
          ISW(1,0)=ISWCU(1,0)
          ISW(2,0)=ISWCU(2,0)
          DO  129  IO=ISW(1,0),ISW(2,0)
             ISW(1,IO)=ISWCU(1,IO)
             ISW(2,IO)=ISWCU(2,IO)
  129     CONTINUE
          ENDIF
       IF (Z2.EQ.47.) THEN
          ISW(1,0)=ISWAG(1,0)
          ISW(2,0)=ISWAG(2,0)
          DO  147  IO=ISW(1,0),ISW(2,0)
             ISW(1,IO)=ISWAG(1,IO)
             ISW(2,IO)=ISWAG(2,IO)
  147     CONTINUE
          ENDIF
C CONSTANTS
       DELD=1./DEL
       TEMPD=1./TEMP
       PILA=10.*PILAM
       RC=ROC*(A1**P1+A2**P1)
       RCMOD=RC
       IF (ICALC.EQ.120.OR.MOD(ICALC,5).EQ.2) THEN
          RCMOD=1.225*(A1**P1+A2**P1)+DD
          ENDIF
       R1=1.28*A1**P1-0.76+0.8*A1**PM1
       C1=R1-1./R1
       R2=1.28*A2**P1-0.76+0.8*A2**PM1
       C2=R2-1./R2
C REACTION PROBABILITIES
       DO  204  IP=1,IPMAX
          A3=A3P(IP)
          A4=A12-A3
          Z3=Z3P(IP)
          Z4=Z12-Z3
          QC=(Z3*Z4-Z1*Z2)*1.44/RC
C MODIFIED VALUES RC AND QC
          P(IP)=EXP((QGG(IP)-QC)*TEMPD)
          IF (ICALC.EQ.120.OR.MOD(ICALC,5).EQ.2) THEN
             QCMOD=(Z3*Z4-Z1*Z2)*1.44/RCMOD
             PMOD(IP)=EXP((QGG(IP)-QCMOD)*TEMPD)
```

```
            ENDIF
C LIMITATION OF ANGULAR MOMENTA
         AT=ABS(A4-A2)
         ZT=ABS(Z4-Z2)
         IF (AT.GE.4.) THEN
            RT=1.28*AT**P1-0.76+0.8*AT**PM1
            CT=RT-1./RT
         ELSE
            CT=ROT*AT**P1
         ENDIF
         IF (A2.LE.A4) THEN
            AREDI=A2*AT/(A2+AT)
            CO=C2+CT
            ALCR(IP)=SQRT(CL*AREDI*CO*CO*
     *         (PI4*GAM*C2*CT-E1*Z2*ZT/CO))-0.5
            ALIM(IP)=A1/AT*ALCR(IP)
         ELSE
            AREDI=A1*AT/(A1+AT)
            CO=C1+CT
            ALCR(IP)=SQRT(CL*AREDI*CO*CO*
     *         (PI4*GAM*C1*CT-E1*Z1*ZT/CO))-0.5
            ALIM(IP)=A2/AT*ALCR(IP)
         ENDIF
C THE TRANSMISSION COEFFICIENT, ORIGINAL FORM (TL) AND
C       THE TRANSMISSION COEFFICIENT,MODIFIED FORM (TLRES)
      IF (ICALC.EQ.100.OR.MOD(ICALC,5).EQ.3) THEN
C - - - - WI, SI, FI ,FIWI, FISI
         DO  201  LL=1,LMAX
            TL(LL)=1./(1.+EXP((LL-ALIM(IP))*DELD))
            SIGL(LL,IP)=TL(LL)*P(IP)
            TLRES(LL)=0.
            SIGNOR(LL,IP)=0.
            SUM(LL)=0.
  201    CONTINUE
      ELSE
C - - - - W1, W2, S1, S2, FIF1, FIF1W1, FIF2W2, FIF1S1, FIF2S2
         DO  202  LL=1,LMAX
            TL(LL)=1./(1.+EXP((LL-ALIM(IP))*DELD))
            SIGL(LL,IP)=TL(LL)*P(IP)
            TLRES(LL)=0.
```

```
            SIGNOR(LL,IP)=0.
            SUM(LL)=0.
  202     CONTINUE
          IF (Z3.LE.1.0*IPIN.AND.A3.NE.0.) THEN
            DO  203  LL=1,LFIN
              TLRES(LL)=1./(1.+EXP((LL-LCRF)*DELD))
              SIGNOR(LL,IP)=TLRES(LL)
  203       CONTINUE
            ENDIF
        ENDIF
  204 CONTINUE
C MULTIPLICATION BY P(*) OR PMOD(*)
        IF (ICALC.EQ.110.OR.MOD(ICALC,5).EQ.1) THEN
          DO  205  IP=1,IPMAX
          DO  205  LL=1,LMAX
            SIGNOR(LL,IP)=SIGNOR(LL,IP)*P(IP)
  205     CONTINUE
          ENDIF
        IF (ICALC.EQ.120.OR.MOD(ICALC,5).EQ.2) THEN
          DO  206  IP=1,IPMAX
          DO  206  LL=1,LMAX
            SIGNOR(LL,IP)=SIGNOR(LL,IP)*PMOD(IP)
  206     CONTINUE
          ENDIF
        IF (ICALC.EQ.100.OR.MOD(ICALC,5).EQ.3) THEN
C - - - - - WI, SI, FI ,FIWI, FISI - - - - - -
C THE EXTENDED SUM RULE CONDITION
          DO  207  LL=1,LMAX
            SUM(LL)=0.
            DO  207 IP=1,IPMAX
              SUM(LL)=SUM(LL)+SIGL(LL,IP)
  207     CONTINUE
C NORMALIZED REACTION PROBABILTIES
          DO  209  IP=1,IPMAX
            SM=0.
            DO  208  LL=1,LMAX
              SIGL(LL,IP)=PILA*ZLP(LL)/SUM(LL)*SIGL(LL,IP)
              SM=SM+SIGL(LL,IP)
  208       CONTINUE
C THE ORIGINAL PARTIAL CROSS-SECTIONS
```

```
            SIG(IP)=SM
            SIGRES(IP)=0.
            SIGTOT(IP)=0.
  209     CONTINUE
      ELSE
C - - - - - W1, W2, S1, S2, FIF1, FIF1W1, FIF2W2, FIF1S1, FIF2S2
C THE EXTENDED SUM RULE CONDITION
      DO  210  LL=1,LMAX
         SUM(LL)=0.
         DO  210  IP=1,IPMAX
            SUM(LL)=SUM(LL)+SIGL(LL,IP)+SIGNOR(LL,IP)
  210 CONTINUE
C NORMALIZED REACTION PROBABILTIES
      DO  212  IP=1,IPMAX
         SM=0.
         SMRES=0.
         DO  211  LL=1,LMAX
C THE NEW PARTIAL CROSS-SECTIONS
            IF (LL.LE.LFIN) THEN
               SIGNOR(LL,IP)=PILA*ZLP(LL)/SUM(LL)*SIGNOR(LL,IP)
            ELSE
               SIGNOR(LL,IP)=0.
            ENDIF
            SMRES=SMRES+SIGNOR(LL,IP)
C THE ORIGINAL PARTIAL CROSS-SECTIONS
            SIGL(LL,IP)=PILA*ZLP(LL)/SUM(LL)*SIGL(LL,IP)
            SM=SM+SIGL(LL,IP)
  211     CONTINUE
C THE REACTION CROSS SECTION FOR CHAN.(IP) ORIGINAL PART
         SIG(IP)=SM
C THE REACTION CROSS SECTION FOR CHAN.(IP) NEW PART
         SIGRES(IP)=SMRES
C THE TOTAL REACTION CROSS SECTION FOR CHAN.(IP)
         SIGTOT(IP)=SMRES+SM
  212     CONTINUE
      ENDIF
C SUM OF ISOTOPES: THE ORIGINAL CROSS-SECTION PART,
C        THE NEW PART OF THE REACTION CROSS SECTION, AND
C        THE TOTAL REACTION CROSS SECTION FOR EACH ELEMENT
C - - - - - FOR LITHIUM-TITAN EXCEPT ELEMENT NUMBER 11
```

```
      IF (Z2.EQ.22.) THEN
         SIGRES(11)=0.
         SIG(11)=0.
         ENDIF
C - - - - - IF PARAMETER FIT REQUIRED - - - - - -
      IF (ICALC/10.EQ.10) THEN
C - - - - - - FI, FIWI, FISI
      DO  213  IO=ISW(1,0),ISW(2,0)
         WILORG(IO)=0.
         DO  213  I=0,ISW(2,IO)-1
            WILORG(IO)=WILORG(IO)+SIG(ISW(1,IO)+I)
  213 CONTINUE
      DO  214  IP=1,MPT
         I=XSEC(IP)
         SECFI(IP)=WILORG(I)
  214 CONTINUE
      ENDIF
      IF (ICALC/10.GE.11) THEN
C - - - - - - FIF1, FIF1W1, FIF1S1, FIF2, FIF2W2, FIF2S2
      DO  215  IO=ISW(1,0),ISW(2,0)
         WILORG(IO)=0.
         WILRES(IO)=0.
         WILSUM(IO)=0.
         DO  215  I=0,ISW(2,IO)-1
            WILORG(IO)=WILORG(IO)+SIG(ISW(1,IO)+I)
            WILRES(IO)=WILRES(IO)+SIGRES(ISW(1,IO)+I)
            WILSUM(IO)=WILSUM(IO)+
     +                 SIG(ISW(1,IO)+I)+SIGRES(ISW(1,IO)+I)
  215 CONTINUE
      DO  216  IP=1,MPT
         I=XSEC(IP)
         SECFI(IP)=WILSUM(I)
  216 CONTINUE
      ENDIF
C - - - - - WI, W1, W2 - - - - - -
      IF (ICALC.GE.1.AND.ICALC.LE.3) THEN
      DO  217  IO=ISW(1,0),ISW(2,0)
         WILORG(IO)=0.
         WILRES(IO)=0.
         WILSUM(IO)=0.
```

```
      DO  217  I=0,ISW(2,IO)-1
         WILORG(IO)=WILORG(IO)+SIG(ISW(1,IO)+I)
         WILRES(IO)=WILRES(IO)+SIGRES(ISW(1,IO)+I)
         WILSUM(IO)=WILSUM(IO)+
     +           SIG(ISW(1,IO)+I)+SIGRES(ISW(1,IO)+I)
  217 CONTINUE
      DO  218  IP=1,MPT
         I=XSEC(IP)
         SECOR(IP)=WILORG(I)
         SECRE(IP)=WILRES(I)
         SECTO(IP)=WILSUM(I)
  218 CONTINUE
      ENDIF
C - - - - - SI, S1, S2 - - - - - -
      IF (ICALC.GE.6.AND.ICALC.LE.8) THEN
      DO  219  I=ISW(1,0),ISW(2,0)
         WILORG(I)=SIG(I+2)
         WILRES(I)=SIGRES(I+2)
         WILSUM(I)=WILORG(I)+WILRES(I)
  219 CONTINUE
      DO  220  IP=1,MPT
         I=XSEC(IP)
         SECOR(IP)=WILORG(I)
         SECRE(IP)=WILRES(I)
         SECTO(IP)=WILSUM(I)
  220 CONTINUE
      ENDIF
      RETURN
      END
```

2.5 Fit-Subroutine FITEX

The FITEX chisquare minimising subroutine solves the nonlinear least squares problem using a least squares interpolation between variables and functions or the exact gradient of the functions. Called subroutines are: LILESQ (linear least squares problem), INVATA (inversion of A(transposed)*A), FIT1 (one dimensional minimum search). Calling sequence:

```
   ke=0
   m=number of functions, m ge n; n=number of variables, n ge 1
   do 1 i=1,n
   x(i)=starting values of the variables
 1 e(i)=absolute search accuracies for the variables, e(i) ne 0
   w(1)=first step size in units of e(i), if le 1 w(1) = 100 by
        fitex the maximum allowed step size is 2*w(1)
   w(2)=method of approximation, 0 for least squares
        interpolation, 1 for exact gradient of the functions
   iw(1)=number of points to be remembered, if le n iw(1) = n+1
   iw(2)=maximum number of function evaluations, if eq 0
         iw(2)=2iw(1), if iw(2) lt 0 no action except ke = 0
   ja=4+max0(14,(n*(n+5))/2)+(m+n+1)*(iw(1)+1)
 2 w(4)=0.
   do 3 i=1,m
   f(i)=function values at the point x
   if(w(2).eq.0.) go to 3
   w(ja+i+m*(j-1))= df(i)/dx(j) for j=1,n
 3 w(4)=w(4)+f(i)*f(i)
   call fitex(ke,m,n,f,x,e,w,iw)
   if(ke.eq.1) go to 2
   w(3)=error renormalisation factor
   w(4)=minimum quadratic sum of the f(i)
   x=minimum point; f=functions at the minimum point
   ke=error code, ke=0: without errors ke=2: user interrupt;
      returns minimum values without errors. The current point
      is ignored. For normal user interrupt set iw(2)=iw(3).
      ke=3: maximum number of function evaluations
      ke=4: rounding errors
      ke=5: the functions do not depent on x(iw(4))
      ke=6: useless variables in the preparatory calls,
            the labels of the variables are iw(3),iw(4).
```

```
   ke=7: m < n or n < 0 or w(2)*(w(2)-1.) ne 0
w(4+i)=standard errors of the variables.
       The error calculation assumes linear functions.
       the program shows the linearity by the kind of
       prediction iw(3); iw(3)=0: linear prediction;
       iw(3)=1: step size limitation; iw(3)=2: one
       dimensional search; iw(3)=3: random search. The
       errors are correctly calculated if the last n
       iterations were linear, i.e. iw(3)=0.
w(4+n+i)=error enhancements; w(4+n+n+i+(j*(j-1))/2)=
   error correlation between x(i) and x(j) i<j
iw(3): number of function evaluations
iw(4): number of degrees of freedom
working fields:  iw: length 4+k with k = iw(1)
   w: length 4+max(14,(n*(n+5))/2)+(m+n+1)*(k+1)+m*n;
   adresses in iw, 4+l: labels of the quadratic sums;
   adresses in w, 4+i: standard error of x(i),
   4+n+i: error enhancement for x(i), from 4+n+n+1: matrix
   d and error correlations from js+1 matrix s; js = 4 +
   max(14,(n*(n+5))/2) from ja+1: matrix a with ja = js +
   (m+n+1)*(k+1).
   The working fields contain all information for the
   continuation of the search. This allows a search within
   another search just changing the working fields.

C-------------------------------------------------------------
      SUBROUTINE FITEX(KE,M,N,F,X,E,W,IW)
      INTEGER IW(M),KE,M,N,
      INTEGER I,I1,I2,IR,J,J1,J2,J3,JA,JD,JM,JS,K,KV,L,LM,MF
      REAL W(M),E(N),F(M),X(N),EPS,S,T,U,V,A,BIG
      EQUIVALENCE (A,IR)
      DATA EPS/1.E-5/,BIG/1.E+38/
      IF (IW(2).LT.0) GOTO 50
      JD = 4 + N + N
      JS = 4 + MAX0(14,(N*(N+5))/2)
      LM = M + N + 1
      IF (KE.NE.0) GOTO 2
      IF (IW(1).LE.N) IW(1) = N + 1
      IF (IW(2).EQ.0) IW(2) = 2*IW(1)
      IF (W(1).LE.1.) W(1) = 100.
```

```
      IW(3) = 1
      K = IW(1)
      DO 1 L = 1,K
         IW(L+4) = 1 + K - L
    1 W(JS+LM*L) = 1.E+38
      KE = 1
    2 K = IW(1)
      KV = K
      JA = JS + LM* (K+1)
      JM = JS + LM*IW(5) - LM
      J3 = JA - LM
      IF (KE.EQ.2) GOTO 52
      IF (M.LT.N.OR.N.LT.1.OR.W(2)*(W(2)-1.).NE.0.) GOTO 57
      IF (W(4).LE.0.) GOTO 50
      L = IW(K+4)
      IF (W(JS+LM*L).EQ.BIG) KV = L - 1
      DO 3 I = 1,K
         J1 = JS + LM*IW(I+4)
         IF (W(4).LT.W(J1)) GOTO 4
    3 CONTINUE
      GOTO 37
    4 IF ((W(2).EQ.0..AND.I.GT.MAX0(N+1,KV)).OR.
     +   (W(2).EQ.1..AND.I.GT.1)) GOTO 37
      IF (KV.LT.K) KV = KV + 1
      I1 = K + 4
      I2 = K - I
      IF (I2.EQ.0) GOTO 6
      DO 5 J = 1,I2
         I1 = I1 - 1
    5 IW(I1+1) = IW(I1)
      IW(I1) = L
      JM = JS + LM*IW(5) - LM
C   NEW ROW
    6 J1 = JS + LM* (L-1)
      DO 7 I = 1,N
         J1 = J1 + 1
    7 W(J1) = X(I)
      DO 8 I = 1,M
         J1 = J1 + 1
    8 W(J1) = F(I)
```

```
      W(J1+1) = W(4)
C   TEST MAXIMUM NUMBEROF FUNCTION EVALUATIONS
      IF (IW(3).GE.IW(2)) GOTO 53
      IF (N.EQ.1) GOTO 42
C   EXACT GRADIENTS OR END OF PREPARATORY FUNCTION EVALUATIONS
      IF (W(2).EQ.1..OR.IW(3).GT.N+1) GOTO 15
C   PREPARATORY FUNCTION EVALUATIONS
      MF = IW(3)
      IF (MF.EQ.1) GOTO 12
      X(MF-1) = W(3)
      J2 = JS + N
      S = 0.
      DO 9 I = 1,M
         T = F(I) - W(J2+I)
    9 S = S + T*T
      J = 2
      IF (S.LT.EPS*EPS*W(JS+LM)) GOTO 55
      W(3) = S
      J1 = 2 + N + MF
      W(J1) = SQRT(W(3))
      IF (MF.LE.2) GOTO 12
      I1 = N + 1
      DO 11 J = 3,MF
         I2 = J2 + LM* (J-2)
         S = 0.
         DO 10 I = 1,M
   10    S = S + (W(I2+I)-W(J2+I))* (F(I)-W(J2+I))
         IF (ABS(W(J1)*W(I1+J)-ABS(S)).LT.EPS*ABS(S)) GOTO 56
   11 CONTINUE
   12 IF (MF.EQ.N+1) GOTO 15
      W(3) = X(MF)
      X(MF) = X(MF) + W(1)*E(MF)
      GOTO 100
C   END OF PREPARATORY FUNCT. SUM OF INVERSES OF QUADRATIC SUMS
   15 S = 0.
      DO 16 L = 1,KV
         T = W(JS+LM*L)
   16 S = S + 1./ (T*T)
      W(JA) = 1./S
C   CENTRE OF THE VARIABLES AND FUNCTIONS
```

```
      I1 = M + N
      DO 18 I = 1,I1
         J1 = JS
         S = 0.
         DO 17 L = 1,KV
            T = W(J1+LM)
            S = S + W(J1+I)/ (T*T)
   17    J1 = J1 + LM
   18 W(J3+I) = S*W(JA)
      IF (KE.NE.1) GOTO 60
      IF (W(2).EQ.0.) GOTO 20
      J1 = JA - M - 1
      DO 19 I = 1,M
   19 W(J1+I) = F(I)
      GOTO 23
C  MATRIX A
   20 J1 = JA
      DO 22 I = 1,N
         U = W(J3+I)
         DO 22 J = 1,M
            J1 = J1 + 1
            J2 = JS
            S = 0.
            T = W(J3+N+J)
            DO 21 L = 1,KV
               V = W(J2+LM)
               S = S + (W(J2+N+J)-T)* (W(J2+I)-U)/ (V*V)
   21       J2 = J2 + LM
   22 W(J1) = S*W(JA)
      IF (KE.NE.1) GOTO 62
C  LINEAR LEAST SQUARES PROBLEM
   23 CALL LILESQ(M,N,IR,W(JA+1),W(JA-M),W(5),W(N+5))
      IF (IR) 54,24,35
C  MATRIX D
   24 J1 = JD
      DO 26 I = 1,N
         T = W(J3+I)
         DO 26 J = 1,I
            J1 = J1 + 1
            J2 = JS
```

```
               S = 0.
               U = W(J3+J)
               DO 25 L = 1,KV
                  V = W(J2+LM)
                  S = S + (W(J2+I)-T)* (W(J2+J)-U)/ (V*V)
   25          J2 = J2 + LM
   26 W(J1) = S*W(JA)
C  NEW VARIABLES
         IF (W(2).EQ.0.) GOTO 28
         DO 27 I = 1,N
   27 X(I) = W(JM+I) - W(I+4)
         GOTO 31
   28 DO 30 I = 1,N
            I2 = 1
            J1 = JD + (I*I-I)/2
            S = 0.
            DO 29 J = 1,N
               J1 = J1 + I2
               IF (J.GE.I) I2 = J
   29       S = S + W(J1)*W(J+4)
   30 X(I) = W(J3+I) - S
C  TEST OF CONVERGENCE
   31 A = 0.
         DO 32 I = 1,N
            W(I+4) = X(I) - W(JM+I)
   32 A = MAX(A,ABS(W(I+4)/E(I)))
         IF (A.LT.1.) GOTO 50
         IW(4) = 0
         W(3) = 1.
         IF (A.LT.2.*W(1)) GOTO 33
C  STEP SIZE LIMITATION
         IW(4) = 1
         W(3) = 2.*W(1)/A
   33 DO 34 I = 1,N
   34 X(I) = W(JM+I) + W(3)*W(I+4)
         GOTO 100
C  RANDOM PREDICTION
   35 DO 36 I = 1,N
            A = W(J3+I)
   36 X(I) = W(JM+I) + W(1)*E(I)* (MOD(IABS(IR),200)-100)/100.
```

```
      IW(4) = 3
      GOTO 100
C     ONE DIMENSIONAL SEARCH
   37 IF (N.EQ.1) GOTO 43
      IF (IW(3).GE.IW(2)) GOTO 53
      IF (IW(4).EQ.2) GOTO 39
      IW(4) = 2
      DO 38 I = 1,N
   38 W(J3+I) = X(I) - W(JM+I)
      IR = 3
      W(5) = IR
      IR = 20
      W(6) = IR
      W(8) = 0.5
      W(11) = 0.
      W(12) = 0.
      W(13) = 0.
      W(14) = 1.
      W(16) = W(JM+LM)
      W(17) = W(4)
      GOTO 40
   39 W(9) = W(4)
      CALL FIT1(KE,W(5),W(8))
   40 DO 41 I = 1,N
   41 X(I) = W(JM+I) + W(8)*W(J3+I)
      IF (KE.EQ.3) KE = 2
      IF (KE.EQ.2) GOTO 53
      KE = 1
      W(3) = W(8)
      GOTO 100
C     ONLY ONE VARIABLE X
   42 IF (IW(3).GT.1) GOTO 43
      KE = 0
      W(10) = W(1)*E(1)
      W(11) = E(1)
      W(12) = 0.
   43 IR = IW(2)
      W(6) = A
      W(8) = X(1)
      W(9) = W(4)
```

```
      CALL FIT1(KE,W(5),W(8))
      IW(4) = 2
      X(1) = W(8)
      IF (KE.EQ.1) GOTO 100
      IF (KE.GT.0) KE = KE + 1
      W(3) = 0.
      W(5) = 0.
      IF (W(6).NE.0.) GOTO 74
      W(5) = SQRT(ABS((W(13)-W(15))/ ((W(16)-W(17))/
     /(W(13)-W(14))-(W(17)-W(18))/ (W(14)-W(15)))))
      W(6) = 1.
      W(7) = 1.
      GOTO 71
C   END OF SEARCH
   50 KE = 0
      IF (W(4).EQ.0. .OR. IW(2).LT.0) GOTO 100
      GOTO 52
C   ERROR CODE DEFINITION
   57 KE = KE + 1
   56 KE = KE + 1
   55 KE = KE + 1
   54 KE = KE + 1
   53 KE = KE + 1
      KE = KE + 1
   52 DO 51 I = 1,N
   51 W(I+4) = 0.
      W(3) = 0.
      IF (KE*(KE-3).NE.0.OR.(KE.EQ.3.AND.(W(2).EQ.1..OR.
     @    (W(3).EQ.0..AND.IW(3).LE.N)))) GOTO 74
C   COMPUTE THE ERRORS OF THE VARIABLES; RESTORE MATRIX G
      IF (W(2).EQ.0.) GOTO 15
      J1 = JA
      I1 = N + 1
      DO 45 I = 2,I1
         IF (I.GT.M) GOTO 45
         DO 44 J = I,M
   44    W(J1+J) = 0.
   45 J1 = J1 + M
      DO 49 I = 1,N
         DO 46 I1 = I,N
```

```
          A = W(4+N+I1)
          IF (IR.EQ.I) GOTO 47
   46     CONTINUE
   47     IF (I1.EQ.I) GOTO 49
          J1 = JA + M* (I-1)
          J2 = JA + M* (I1-1)
          W(4+N+I1) = W(4+N+I)
          DO 48 J = 1,N
             A = W(J1+J)
             W(J1+J) = W(J2+J)
   48     W(J2+J) = A
   49  CONTINUE
       GOTO 66
C   INVERSE OF MATRIX D
   60  T = SQRT(W(JA))
       J1 = JA
       DO 61 I = 1,N
          S = W(J3+I)
          J2 = JS + I - LM
          DO 61 L = 1,KV
             J1 = J1 + 1
   61  W(J1) = T*(W(J2+L*LM)-S)/W(JS+L*LM)
       CALL INVATA(KV,N,IR,W(JA+1),W(JD+1),X)
       IF (IR) 74,20,74
C   MATRIX G = A*INVERSE OF D
   62  DO 65 L = 1,M
          J1 = L + JA - M
          DO 64 I = 1,N
             I1 = JD + (I*I-I)/2
             I2 = 1
             S = 0.
             DO 63 J = 1,N
                I1 = I1 + I2
                IF (J.GE.I) I2 = J
   63           S = S + W(I1)*W(J1+J*M)
   64     X(I) = S
          DO 65 J = 1,N
   65  W(J1+J*M) = X(J)
C   DIAGONAL ELEMENTS OF G(T)*G
   66  J1 = JA
```

```
      DO 68 I = 1,N
         S = 0.
         DO 67 L = 1,M
            J1 = J1 + 1
   67    S = S + W(J1)*W(J1)
   68 W(4+N+I) = SQRT(S)
C  STANDARD ERRORS AND ERROR CORRELATIONS
      CALL INVATA(M,N,IR,W(JA+1),W(JD+1),X)
      IF (IR.NE.0) GOTO 74
      DO 69 I = 1,N
         W(I+4) = SQRT(W(JD+ (I*I+I)/2))
   69 W(4+N+I) = W(I+4)*W(4+N+I)
      J1 = JD
      DO 70 I = 1,N
         DO 70 J = 1,I
            J1 = J1 + 1
   70 W(J1) = W(J1)/ (W(I+4)*W(J+4))
C  ERROR RENORMALISATION FACTOR
   71 S = 0.
      DO 72 I = 1,M
   72 S = S + W(JM+N+I)
      W(3) = SQRT(ABS(W(JM+LM)-S*S/M)/MAX0(M-N-1,1))
      DO 73 I = 1,N
   73 W(I+4) = W(I+4)*W(3)
C  RESTORE OPTIMUM VALUES TO X AND F
   74 IW(4) = M - N - 1
      IF ((KE-5)* (KE-6).NE.0) GOTO 75
      IW(3) = J - 2
      IW(4) = MF - 1
   75 DO 76 I = 1,N
   76 X(I) = W(JM+I)
      DO 77 I = 1,M
   77 F(I) = W(JM+N+I)
      W(4) = W(JM+LM)
  100 IF (KE.EQ.1) IW(3) = IW(3) + 1
      RETURN
      END
C-------------------------------------------------------------
```

```
      SUBROUTINE FIT1(KE,V,W)
      INTEGER KE,IV,J,K
      REAL  V(3),W(11)
      IF (KE.EQ.1) GOTO 2
      KE = 1
      V(1) = 1
      V(3) = -1
      W(6) = W(1)
      W(9) = W(2)
    1 W(1) = W(1) + W(3)
      GOTO 12
    2 IF (V(1).GT.2.) GOTO 3
      V(3) = 0.
      W(7) = W(1)
      W(10) = W(2)
      IF (W(2).LE.W(9)) GOTO 1
      V(3) = -1.
      W(1) = W(6) - W(3)
      GOTO 12
    3 IF (V(1).GT.3.) GOTO 5
      W(8) = W(1)
      W(11) = W(2)
      DO 4 J = 1,3
         K = 7 - MOD(J,2)
         IF (W(K).LE.W(K+1)) GOTO 4
         W(1) = W(K)
         W(K) = W(K+1)
         W(K+1) = W(1)
         K = K + 3
         W(1) = W(K)
         W(K) = W(K+1)
         W(K+1) = W(1)
    4 CONTINUE
      V(3) = 0.
      IF (W(9).LT.W(10).AND.W(9).LT.W(11)) V(3) = -1.
      IF (W(11).LT.W(10).AND.W(11).LT.W(9)) V(3) = 1.
      GOTO 9
C   SORT IN THE NEW VALUES OF X AND F
    5 IF (V(3).EQ.0.) GOTO 6
      J = V(3)
```

```
      W(7-J) = W(7)
      W(10-J) = W(10)
      IF ((W(7+J)-W(1))*(W(1)-W(7)).GT.0.) GOTO 7
      W(7) = W(7+J)
      W(10) = W(10+J)
      W(7+J) = W(1)
      W(10+J) = W(2)
      IF (W(2).GE.W(10)) V(3) = 0.
      GOTO 9
    6 J = -1
      IF (W(1).LT.W(7)) J = 1
      IF (W(2).GT.W(10)) GOTO 8
      W(7+J) = W(7)
      W(10+J) = W(10)
    7 W(7) = W(1)
      W(10) = W(2)
      IV = V(3)
      IF (W(2).LE.W(10+IV)) V(3) = 0.
      GOTO 9
    8 W(7-J) = W(1)
      W(10-J) = W(2)
    9 IV = V(3)
      J = 7 + IV
C   ERROR TESTS
      IF (W(6).EQ.W(7) .OR. W(7).EQ.W(8) .OR.
     @    (W(9).EQ.W(10).AND.W(10).EQ.W(11))) GOTO 15
      IF (V(1).GE.V(2)) GOTO 16
      IF (V(3).EQ.0.) GOTO 10
C   STEP SIZE LIMITATION
      W(1) = W(J) + 2.*V(3)* (W(8)-W(6))
      GOTO 12
   10 W(1) = MIN(W(8)-W(7),W(7)-W(6))/(W(8)-W(6))
      IF (W(1).GT.0.1) GOTO 11
      W(1) = .5* (W(6)+W(8))
      GOTO 12
C   PREDICTION OF THE POSITION OF THE MINIMUM
   11 W(1) = ((W(9)-W(10))/ (W(6)-W(7))- (W(10)-W(11))/
     / (W(7)-W(8)))/ (W(6)-W(8))
      W(1) = .5*(W(6)+W(8)+(W(11)-W(9))/ (W(1)* (W(6)-W(8))))
C   TEST OF CONVERGENCE
```

```
      W(2) = ABS(W(1)-W(J))
      IF (W(2).LT.ABS(W(4)).OR.W(2).LT.ABS(W(5)*W(J))) GOTO 13
   12 V(1) = V(1) + 1.
      RETURN
   13 KE = 0
   14 IV = V(3)
      W(1) = W(7+IV)
      W(2) = W(10+IV)
      RETURN
   15 KE = KE + 1
   16 KE = KE + 1
      GOTO 14
      END
C-------------------------------------------------------------
      SUBROUTINE INVATA(M,N,IR,A,D,VP)
      INTEGER IR,M,N,I,I1,IJ,J,K,L
      REAL  A(M,N),D(N),VP(N),EPS,P,Q,R,S,SIG,T,U,V,C
      DATA EPS/1.E-5/
      IR = N
      IF (M.LT.N.OR.N.LT.1) GOTO 19
      DO 1 I = 1,IR
    1 VP(I) = I
C   HOUSEHOLDER LOOP
      K = 0
    2 K = K + 1
C   PIVOT ELEMENT
    3 C = 0.
      DO 4 I = K,M
         IF (ABS(A(I,K)).LE.C) GOTO 4
         C = ABS(A(I,K))
         I1 = I
    4 CONTINUE
      IF (C.GT.0.) GOTO 8
      IR = IR - 1
      IF (K.GT.IR) GOTO 13
C   SET UP THE PERMUTATION VECTOR IP AND
C         PERMUTE THE COLUMNS OF MATRIX A
      L = VP(K)
      DO 5 J = K,IR
    5 VP(J) = VP(J+1)
```

```
      VP(IR+1) = L
      DO 7 I = 1,M
         C = A(I,K)
         DO 6 J = K,IR
    6    A(I,J) = A(I,J+1)
    7 A(I,IR+1) = C
      GOTO 3
C  ROTATION OF THE LOWER COLUMN FRAGMENTS OF A(K)
    8 DO 9 J = K,IR
         C = A(K,J)
         A(K,J) = A(I1,J)
    9 A(I1,J) = C
      S = A(K,K)
      V = 0.
      DO 10 I = K,M
         U = A(I,K)/S
   10 V = V + U*U
      V = 1./SQRT(V)
      SIG = S/V
      U = S + SIG
      A(K,K) = -SIG
      IF (K.GE.IR) GOTO 13
      L = K + 1
      DO 12 J = L,IR
         S = V*A(K,J)
         P = ABS(S)
         DO 11 I = L,M
            R = (A(I,K)/SIG)*A(I,J)
            S = S + R
   11    P = P + ABS(R)
         IF (ABS(S).LE.EPS*P) S = 0.
         T = (A(K,J)+S)/U
         IF (ABS(T).LE.EPS*ABS(S/U)) T = 0.
         A(K,J) = -S
         DO 12 I = L,M
            Q = A(I,J)
            P = T*A(I,K)
            R = Q - P
            IF (ABS(R).LE.EPS*ABS(P)) R = 0.
   12 A(I,J) = R
```

```
      GOTO 2
C   END OF HOUSEHOLDER LOOP
   13 IF (IR.EQ.0) GOTO 20
C   INVERSE OF THE TRIANGULAR MATRIX R STORED IN D
      IJ = 0
      DO 16 K = 1,IR
         D(IJ+K) = 1./A(K,K)
         IF (K.EQ.1) GOTO 16
         I = K
         DO 15 L = 2,K
            I1 = I
            I = I - 1
            S = 0.
            DO 14 J = I1,K
   14       S = S + A(I,J)*D(IJ+J)
   15    D(IJ+I) = -S/A(I,I)
   16 IJ = IJ + K
C   INVERSE OF THE PRODUCT MATRIX
      IJ = 0
      DO 18 J = 1,IR
         DO 18 I = 1,J
            IJ = IJ + 1
            I1 = IJ
            L = J - I
            S = 0.
            DO 17 K = J,IR
               S = S + D(I1)*D(I1+L)
   17       I1 = I1 + K
   18 D(IJ) = S
      GOTO 20
   19 IR = -2
   20 IF (IR.EQ.0) IR = -1
      IF (IR.EQ.N) IR = 0
      RETURN
      END
C--------------------------------------------------------------
      SUBROUTINE LILESQ(M,N,IER,A,B,X,VP)
      INTEGER IER,M,N,I,IP,J,K,L,L1,L2
      REAL C,DELTA,EPS,P,Q,R,S,SIG,T,U,V,W
      REAL A(M,N),B(M),VP(N),X(N)
```

```
      DATA EPS/1.E-5/
      IER = 0
      IF (M.LT.N.OR.N.LT.1) GOTO 19
      DO 1 J = 1,N
    1 VP(J) = J
C   ROTATION LOOP
      DO 10 K = 1,N
C   PIVOT ELEMENT
         U = 0.
         DO 4 J = K,N
            C = 0.
            DO 2 I = K,M
               IF (ABS(A(I,J)).LE.ABS(C)) GOTO 2
               L2 = I
               C = A(I,J)
    2       CONTINUE
            IF (C.EQ.0.) GOTO 4
            S = 0.
            T = 0.
            DO 3 I = K,M
               V = A(I,J)/C
               S = S + V*V
    3       T = T + V*B(I)
            IF (U.GE.T* (T/S)) GOTO 4
            U = T* (T/S)
            SIG = C*SQRT(S)
            W = T
            L = J
            L1 = L2
    4    CONTINUE
         IF (U.EQ.0.) GOTO 11
C   PERMUTE A(K) AND B(K)
         I = VP(L)
         VP(L) = VP(K)
         VP(K) = I
         DO 5 I = 1,M
            C = A(I,L)
            A(I,L) = A(I,K)
    5    A(I,K) = C
         C = B(K)
```

```
      B(K) = B(L1)
      B(L1) = C
      DO 22 J = K,N
         C = A(K,J)
         A(K,J) = A(L1,J)
   22 A(L1,J) = C
C  ROTATION OF THE LOWER COLUMN FRAGMENT OF A(K) AND B(K)
      U = SIG + A(K,K)
      V = A(K,K)/SIG
      DELTA = (B(K)+V*W)/U
      A(K,K) = -SIG
      B(K) = -V*W
      L = K + 1
      IF (L.GT.M) GOTO 10
      IF (K.GE.N) GOTO 8
      DO 7 J = L,N
         S = V*A(K,J)
         P = ABS(S)
         DO 6 I = L,M
            R = A(I,K)/SIG*A(I,J)
            S = S + R
    6    P = P + ABS(R)
         IF (ABS(S).LE.EPS*P) S = 0.
         T = (A(K,J)+S)/U
         IF (ABS(T).LE.EPS*ABS(S/U)) T = 0.
         A(K,J) = -S
         DO 7 I = L,M
            Q = A(I,J)
            P = T*A(I,K)
            R = Q - P
            IF (ABS(R).LE.EPS*ABS(P)) R = 0.
    7 A(I,J) = R
    8 DO 9 I = L,M
    9 B(I) = B(I) - DELTA*A(I,K)
   10 CONTINUE
C  END OF ROTATION LOOP
      K = N
      GOTO 12
   11 K = K - 1
```

```
      IER = K
C   SQUARE OF THE EUCLIDEAN NORM
   12 S = 0.
      L = K + 1
      IF (K.EQ.M) GOTO 14
      DO 13 I = L,M
   13 S = S + B(I)*B(I)
   14 A(2,1) = S
      IF (K.EQ.N) GOTO 16
C   COMPONENTS OF X WHICH DO NOT REDUCE THE EUCLIDEAN NORM
      DO 15 I = L,N
         DO 15 J = L,N
            IP = VP(J)
            X(IP) = 0.
   15 CONTINUE
      IF (K.EQ.0) GOTO 20
C   COMPUTATION OF X
   16 CONTINUE
      IP = VP(K)
      X(IP) = B(K)/A(K,K)
      IF (K.EQ.1) GOTO 21
      DO 18 J = 2,K
         L = K + 2 - J
         S = B(L-1)
         DO 17 I = L,K
            IP = VP(I)
   17    S = S - A(L-1,I)*X(IP)
         IP = VP(L-1)
         X(IP) = S/A(L-1,L-1)
   18 CONTINUE
      GOTO 21
C   ERROR CODE
   19 IER = IER - 1
   20 IER = IER - 1
   21 RETURN
      END
```

3. USE OF IMFREM

3.1 IDENTIFIERS FOR THE AVAILABLE CALCULATIONS

'FI'	=	original model; FITEX-fit
'F1'	=	extended model; FITEX-fit
'F2'	=	extended model; modified; FITEX-fit
'WI'	=	original model; total reaction cross section
'W1'	=	extended model; total reaction cross section
'W2'	=	extended model; modified; total reaction cross section
'SI'	=	original model; partial cross section
'S1'	=	extended model; partial cross section
'S2'	=	extended model; modified; partial cross section

3.2 IDENTIFIERS FOR POSSIBLE CALCULATIONS

'W1'	for	W1	only	
'W2'	for	W2	only	
'WI'	for	WI	only	
'S1'	for	S1	only	
'S2'	for	S2	only	
'FI'	for	FI	only	
'SI'	for	SI	only	
'FIWI'	for	FI	and	WI
'FISI'	for	FI	and	SI
'FIF1'	for	(FI + F1)		
'FIF1W1'	for	(FI + F1)	and	W1
'FIF1S1'	for	(FI + F1)	and	S1
'FIF2'	for	(FI + F2)		
'FIF2W2'	for	(FI + F2)	and	W2
'FIF2S2'	for	(FI + F2)	and	S2

3.3 DESCRIPTION OF INPUT VALUES IN DETAIL

Remark: Each input line is composed of two records of the "G.SYSIN stream": Leading a comment line, which text will not be used, and after that the line with the character strings, or the real or integer values.

ZEROTH LINE

Valid for all options

```
* CALCULATION & PROJECTILE-TARGET CHARACTER STRING
      'FI'            'LIAG'
```

F I R S T   L I N E

Valid for all options

```
* MASS & PROTONS OF PROJECTILE AND TARGET, LAB. ENERGY
  6.  3.  108.  47.  156.
```

A1 = mass of the projectile
Z1 = charge or number of protons of the projectile
A2 = mass of the target
Z2 = charge or number of protons of the target
ELAB = incident energy

S E C O N D   L I N E

Valid for **FI, WI, SI**

```
* LMAX, MAX. OF REACTION CHANNELS
  56  75
```

LMAX = maximum value of l_{max}
IPMAX = maximum number of reaction channels

S E C O N D   L I N E

Valid for **FIF1, FIF2, W1, W2, S1, S2**

```
* LMAX, LMAX 2ND TERM, MAX.REACT.CHAN. 1ST, 2ND, CRIT.ANG.MOM.
  58  58  75  64  51
```

LMAX = maximum value of l_{max}, the angular momentum confining the interval of the sum in the l-space
LFIN = maximum value of l_{max} for the second term

IPMAX = maximum number of reaction channels
IPIN = maximum number of reaction channels of interest for the second term
LCRF = value of the critical angular momentum for fusion with dissipation, l_{cr}^{dyn} (calculated with a dynamical model)

T H I R D   L I N E

Valid for **FI, FIF1, WI, W1, SI, S1**

```
* INITIAL OR FIXED VALUES FOR FIT PARAMETERS (TEMP,ROC,DEL)
  3.  1.5  3.0
```

TEMP = temperature
ROC = radius R_{oc}
DEL = Δl

T H I R D   L I N E

Valid for **FIF2, W2, S2**

```
* INITIAL OR FIXED VALUES FOR FIT PARAMETERS (TEMP,ROC,DD,DEL)
  2.11  1.5   6.7  3.0
```

TEMP = temperature
ROC = radius R_{oc}
DD = additional term for the modified R_c
DEL = Δl

Q G G - D A T A

Valid for **SI, S1, S2** "READ(2,*)"

Calculated Q_{gg} values (Q-values of the ground state)

```
          DS$LIAG
            27
        0.   0.   14.674
        0.   1.    7.022
         .    .    .
```

Leading a comment line, second the number of data points, after that beginning with the third line the data points.

Q G G - D A T A

Valid for **FI, FIF1, FIF2, WI, W1, W2** "READ(3,*)"

Calculated Q_{gg} values (Q-values of the ground state)

```
        D$LIAG
          75
     0.    0.    14.674
     0.    1.     7.022
      .     .      .
```

Leading a comment line, second the number of data points, after that beginning with the third line the data points.

E X P E R I M E N T A L   D A T A

Valid for all options "READ(4,*)"

Experimental reaction cross-section data

Leading a comment line, second the number of data points, after that beginning with the third line the data points.

```
     DE$LIAG
         13
      2.,  519.7
      4.,    2.1
       .      .
```

3.4 EXAMPLE WITH THE FULL SET OF CONTROL STATEMENTS

```
//IAK964X  JOB (0964,145,POC1A),OEHLSCHLAEGER,NOTIFY=IAK964,
//     MSGCLASS=H,REGION=2000K,MSGLEVEL=(2,0)
//*MAIN LINES=19
//  EXEC  F7C,PARM.C='DEBUG(UNDEF,SUBCHK),LINECOUNT(64)'
//* C.SYSPRINT DD DUMMY
//C.SYSIN DD DISP=SHR,DSN=IAK964.ILIANA.FORT(IMFREM)
//  EXEC  F7CLG,PARM.C='DEBUG(UNDEF,SUBCHK),LINECOUNT(64)'
//C.SYSPRINT DD DUMMY
//C.SYSIN DD DISP=SHR,DSN=IAK964.ILIANA.FORT(FITFUN)
//         DD DISP=SHR,DSN=IAK964.ILIANA.FORT(FITEX)
//L.SYSPRINT DD DUMMY
//L.SYSIN DD *
 ENTRY IMFREM
//G.FT02F001 DD DISP=SHR,DSN=IAK964.ILIANA.FORT(DS$LIAG)
//G.FT03F001 DD DISP=SHR,DSN=IAK964.ILIANA.FORT(D$LIAG)
//G.FT04F001 DD DISP=SHR,DSN=IAK964.ILIANA.FORT(DE$LIAG)
//G.FT06F001 DD SYSOUT=*
//G.SYSIN DD * * *
* CALCULATION & PROJECTILE-TARGET CHARACTER STRING
      'FIF1W1'           'LIAG'
* MASS & PROTONS OF PROJECTILE AND TARGET, LAB. ENERGY
  6.  3.  108.  47.  156.
* LMAX, LMAX 2ND TERM, MAX.REACT.CHAN. 1ST, 2ND, CR.ANG.MOM.
  58  58  75  64  46
* INITIAL OR FIXED VALUES FOR FIT PARAMETERS (TEMP, ROC, DEL)
  3.3   1.1   3.0
//
```

3.5 OUTPUT OF THE EXAMPLE

```
=======================
START OF PROGRAM IMFREM
======================================================
VERSION FIF1W1 WITH PROJECTILE-TARGET COMBINATION LIAG
======================================================
I C A L C = 111
INPUT VALUES:
A1= 6.0000E+00 Z1= 3.0000E+00 A2= 1.0800E+02 Z2= 4.7000E+01
ELAB= 1.5600E+02
LMAX=  58 LFIN=  58 IPMAX=  75 IPIN=  64 LCRF=  51
TEMP= 1.000 ROC= 1.000 DEL= 3.000
EXPERIMENTAL DATA:
     NUMBER      CANAL            SECTION           ERROR
        1    1.0000000E+00    1.0883000E+03    2.1765999E+02
        2    2.0000000E+00    2.3600000E+02    4.7199997E+01
        3    4.0000000E+00    2.1000004E+00    4.2000002E-01
        4    5.0000000E+00    1.2299995E+00    2.4599987E-01
        5    6.0000000E+00    1.0010004E+00    2.0020002E-01
        6    7.0000000E+00    2.5999999E-01    5.1999994E-02
        7    8.0000000E+00    1.5300000E-01    3.0599996E-02
        8    9.0000000E+00    5.0000001E-02    9.9999979E-03
        9    1.0000000E+01    4.8000000E-02    9.5999986E-03
       10    1.1000000E+01    2.4999999E-02    4.9999990E-03
       11    1.2000000E+01    2.4999999E-02    4.9999990E-03
       12    1.3000000E+01    2.6000001E-02    5.1999986E-03
       13    1.4000000E+01    1.5000001E-02    2.9999998E-03
       14    1.5000000E+01    1.3000000E-02    2.5999998E-03
ADDITIONAL CONSTANTS:
GAM= 9.24497E-01 ARED= 5.68421E+00 LAM= 1.57730E-01
RC= 0.65793E+01 ROT= 0.764 ECM= 1.47789E+02
 1 ITERAT  W(4) = 3.5000E+02  IW(4) =  0  W(3) = 0.0000E+00
     FIT PARAMETERS
          1.00000E+00     1.00000E+00
35 ITERAT  W(4) = 7.9763E+01  IW(4) = 11  W(3) = 2.3914E+00
     FIT PARAMETERS
          3.77833E+00     1.46702E+00
     CONVERGENCE WITH RETURN CODE 0
     FINAL PARAMETERS AT THE END OF SEARCH
```

NUMBER	VALUE	ERROR
1	3.77833E+00	1.84661E-01
2	1.46702E+00	5.85798E-02

"W1"

CAN	Z3	A3	QGG	P	LCR	LIM
1	0.0	0.0	1.46740E+1	9.19710E+3	1.95675E+1	1.95675E+1
2	0.0	1.0	7.02200E+0	1.29325E+3	1.61748E+1	1.94097E+1
3	1.0	1.0	6.50900E+0	1.75751E+2	1.76610E+1	2.11932E+1
4	1.0	2.0	1.25300E+0	4.56772E+1	1.41228E+1	2.11843E+1
5	1.0	3.0	-2.25100E+0	1.86023E+1	1.11444E+1	2.22889E+1
6	2.0	3.0	1.05900E+0	7.26909E+0	1.23823E+1	2.47646E+1
7	2.0	4.0	1.19850E+1	1.19666E+2	9.00996E+0	2.70299E+1
8	2.0	5.0	3.93700E+0	1.52025E+1	5.13100E+0	3.07860E+1
9	3.0	5.0	1.32000E-1	1.03442E+0	5.93987E+0	3.56392E+1
10	4.0	8.0	8.98400E+0	1.94871E+0	1.31913E+0	7.12330E+1
11	4.0	9.0	3.52200E+0	4.80411E-1	1.89779E+0	6.83205E+1
12	5.0	9.0	-2.14000E-1	3.87412E-2	1.78254E+0	6.41715E+1
13	5.0	10.0	1.42100E+0	5.89132E-2	2.27070E+0	6.13090E+1
14	5.0	11.0	3.73500E+0	1.06623E-1	2.86632E+0	6.19125E+1
15	5.0	12.0	-1.34000E-1	3.95439E-2	3.41975E+0	6.15555E+1
16	6.0	11.0	3.88600E+0	2.51324E-2	2.74363E+0	5.92625E+1
17	6.0	12.0	1.31470E+1	2.69990E-1	3.29862E+0	5.93751E+1
18	6.0	13.0	1.15900E+1	1.81131E-1	3.81715E+0	5.88932E+1
19	6.0	14.0	1.02820E+1	1.29527E-1	4.30431E+0	5.81081E+1
20	6.0	15.0	4.35300E+0	2.83289E-2	4.76425E+0	5.71709E+1
21	7.0	13.0	5.50700E+0	9.31828E-3	3.69397E+0	5.69927E+1
22	7.0	14.0	9.50900E+0	2.59965E-2	4.18280E+0	5.64678E+1
23	7.0	15.0	1.16580E+1	4.51007E-2	4.64422E+0	5.57307E+1
24	7.0	16.0	7.11700E+0	1.40796E-2	5.08173E+0	5.48826E+1
25	8.0	16.0	1.70490E+1	4.74330E-2	4.96047E+0	5.35731E+1
26	8.0	17.0	1.48030E+1	2.66694E-2	5.37838E+0	5.28059E+1
27	8.0	18.0	1.39390E+1	2.13706E-2	5.77748E+0	5.19973E+1
28	8.0	19.0	1.07900E+1	9.53256E-3	6.15976E+0	5.11734E+1
29	9.0	17.0	8.18200E+0	1.39171E-3	5.25611E+0	5.16055E+1
30	9.0	18.0	1.08740E+1	2.77508E-3	5.65676E+0	5.09109E+1
31	9.0	19.0	1.30150E+1	4.80456E-3	6.04047E+0	5.01823E+1
32	9.0	20.0	1.25500E+1	4.26461E-3	6.40897E+0	4.94406E+1
33	9.0	21.0	1.19800E+1	3.68483E-3	6.76372E+0	4.86988E+1
34	10.0	20.0	1.89900E+1	6.83387E-3	6.28892E+0	4.85145E+1
35	10.0	21.0	1.91430E+1	7.10725E-3	6.64502E+0	4.78441E+1

36	10.0	22.0	2.10720E+1	1.16540E-2	6.98855E+0	4.71727E+1
37	10.0	23.0	1.92980E+1	7.39536E-3	7.32059E+0	4.65072E+1
38	10.0	24.0	1.63940E+1	3.51260E-3	7.64212E+0	4.58527E+1
39	11.0	23.0	2.07590E+1	3.56808E-3	7.20254E+0	4.57573E+1
40	11.0	24.0	2.09390E+1	3.73659E-3	7.52525E+0	4.51515E+1
41	11.0	25.0	2.02340E+1	3.11875E-3	7.83822E+0	4.45541E+1
42	11.0	26.0	1.54580E+1	9.16697E-4	8.14219E+0	4.39678E+1
43	12.0	24.0	2.44420E+1	3.28361E-3	7.40666E+0	4.44399E+1
44	12.0	25.0	2.56850E+1	4.51592E-3	7.72083E+0	4.38868E+1
45	12.0	26.0	2.72570E+1	6.75729E-3	8.02592E+0	4.33400E+1
46	12.0	27.0	2.62620E+1	5.23587E-3	8.32260E+0	4.28019E+1
47	12.0	28.0	2.34540E+1	2.54887E-3	8.61148E+0	4.22745E+1
48	13.0	26.0	2.09420E+1	5.17094E-4	7.90805E+0	4.27035E+1
49	13.0	27.0	2.53980E+1	1.62068E-3	8.20587E+0	4.22016E+1
50	13.0	28.0	2.57070E+1	1.75429E-3	8.49581E+0	4.17067E+1
51	13.0	29.0	2.49260E+1	1.43597E-3	8.77842E+0	4.12204E+1
52	13.0	30.0	2.19580E+1	6.70949E-4	9.05416E+0	4.07437E+1
53	14.0	28.0	2.81240E+1	1.35882E-3	8.37865E+0	4.11315E+1
54	14.0	29.0	3.02710E+1	2.35617E-3	8.66233E+0	4.06753E+1
55	14.0	30.0	3.06590E+1	2.60259E-3	8.93909E+0	4.02259E+1
56	14.0	31.0	2.99520E+1	2.17115E-3	9.20938E+0	3.97845E+1
57	14.0	32.0	2.84020E+1	1.45919E-3	9.47358E+0	3.93518E+1
58	14.0	33.0	2.52850E+1	6.56251E-4	9.73207E+0	3.89283E+1
59	15.0	30.0	2.45520E+1	2.44600E-4	8.82260E+0	3.97017E+1
60	15.0	31.0	2.76010E+1	5.34478E-4	9.09391E+0	3.92857E+1
61	15.0	32.0	2.76010E+1	5.34478E-4	9.35909E+0	3.88762E+1
62	15.0	33.0	2.84930E+1	6.71807E-4	9.61850E+0	3.84740E+1
63	15.0	34.0	2.50900E+1	2.80774E-4	9.87247E+0	3.80795E+1
64	15.0	35.0	2.32300E+1	1.74287E-4	1.01213E+1	3.76931E+1
65	16.0	34.0	3.16050E+1	7.24067E-4	9.75853E+0	3.76400E+1
66	16.0	35.0	3.17530E+1	7.52069E-4	1.00082E+1	3.72721E+1
67	17.0	37.0	2.98320E+1	2.40696E-4	1.03807E+1	3.61649E+1
68	18.0	39.0	3.23430E+1	2.58932E-4	1.07375E+1	3.51410E+1
69	19.0	43.0	3.11750E+1	1.17040E-4	1.15213E+1	3.36298E+1
70	20.0	47.0	3.69190E+1	3.35818E-4	1.22553E+1	3.22823E+1
71	21.0	49.0	3.76060E+1	2.84366E-4	1.25519E+1	3.15256E+1
72	22.0	50.0	4.26160E+1	7.87085E-4	1.26424E+1	3.10312E+1
73	23.0	52.0	4.00100E+1	3.33620E-4	1.29255E+1	3.03469E+1
74	24.0	54.0	4.32700E+1	6.86509E-4	1.32004E+1	2.97009E+1

75	25.0	56.0	4.00740E+1	2.91261E-4	1.34675E+1	2.90899E+1
NR.CANAL		Z	A	SIG(MB)	SIGRES	SIGTOT
1		0.0E+00	0.0E+00	4.17495E+02	0.00000E+00	4.17495E+02
2		0.0E+00	1.0E+00	5.37886E+01	1.73939E+03	1.79318E+03
3		1.0E+00	1.0E+00	8.88978E+00	2.19356E+02	2.28245E+02
4		1.0E+00	2.0E+00	2.20883E+00	5.45765E+01	5.67854E+01
5		1.0E+00	3.0E+00	1.03184E+00	2.15896E+01	2.26214E+01
6		2.0E+00	3.0E+00	5.57986E-01	8.10441E+00	8.66239E+00
7		2.0E+00	4.0E+00	1.38290E+01	1.46081E+02	1.59910E+02
8		2.0E+00	5.0E+00	2.65542E+00	1.73591E+01	2.00146E+01
9		3.0E+00	5.0E+00	2.76379E-01	1.07270E+00	1.34908E+00
10		4.0E+00	8.0E+00	4.93235E+00	2.04440E+00	6.97676E+00
11		4.0E+00	9.0E+00	1.15290E+00	4.81667E-01	1.63457E+00
12		5.0E+00	9.0E+00	8.19743E-02	3.54996E-02	1.17474E-01
13		5.0E+00	1.0E+01	1.18225E-01	5.47213E-02	1.72946E-01
14		5.0E+00	1.1E+01	2.22189E-01	1.00957E-01	3.23145E-01
15		5.0E+00	1.2E+01	7.89589E-02	3.62593E-02	1.15218E-01
16		6.0E+00	1.1E+01	4.46637E-02	2.25266E-02	6.71903E-02
17		6.0E+00	1.2E+01	5.21123E-01	2.61328E-01	7.82451E-01
18		6.0E+00	1.3E+01	3.36451E-01	1.73069E-01	5.09519E-01
19		6.0E+00	1.4E+01	2.27214E-01	1.22426E-01	3.49640E-01
20		6.0E+00	1.5E+01	4.44096E-02	2.54902E-02	6.98999E-02
21		7.0E+00	1.3E+01	1.38037E-02	8.02634E-03	2.18300E-02
22		7.0E+00	1.4E+01	3.82577E-02	2.31484E-02	6.14061E-02
23		7.0E+00	1.5E+01	6.36529E-02	4.08823E-02	1.04535E-01
24		7.0E+00	1.6E+01	1.77823E-02	1.22907E-02	3.00730E-02
25		8.0E+00	1.6E+01	5.48245E-02	4.27556E-02	9.75801E-02
26		8.0E+00	1.7E+01	2.81149E-02	2.35956E-02	5.17105E-02
27		8.0E+00	1.8E+01	2.06817E-02	1.87724E-02	3.94541E-02
28		8.0E+00	1.9E+01	8.29585E-03	8.15760E-03	1.64535E-02
29		9.0E+00	1.7E+01	1.17871E-03	1.11144E-03	2.29015E-03
30		9.0E+00	1.8E+01	2.24681E-03	2.26628E-03	4.51309E-03
31		9.0E+00	1.9E+01	3.69150E-03	3.99401E-03	7.68550E-03
32		9.0E+00	2.0E+01	3.04188E-03	3.53150E-03	6.57337E-03
33		9.0E+00	2.1E+01	2.44075E-03	3.03698E-03	5.47773E-03
34		1.0E+01	2.0E+01	4.51106E-03	5.70935E-03	1.02204E-02
35		1.0E+01	2.1E+01	4.41857E-03	5.94530E-03	1.03639E-02
36		1.0E+01	2.2E+01	6.93199E-03	9.90602E-03	1.68380E-02
37		1.0E+01	2.3E+01	4.08775E-03	6.19426E-03	1.02820E-02
38		1.0E+01	2.4E+01	1.79078E-03	2.87206E-03	4.66284E-03

39	1.1E+01	2.3E+01	1.79431E-03	2.90143E-03	4.69574E-03
40	1.1E+01	2.4E+01	1.78704E-03	3.04299E-03	4.83003E-03
41	1.1E+01	2.5E+01	1.40991E-03	2.52502E-03	3.93493E-03
42	1.1E+01	2.6E+01	3.79231E-04	7.13340E-04	1.09257E-03
43	1.2E+01	2.4E+01	1.46452E-03	2.64810E-03	4.11262E-03
44	1.2E+01	2.5E+01	1.94304E-03	3.67966E-03	5.62270E-03
45	1.2E+01	2.6E+01	2.81461E-03	5.57826E-03	8.39287E-03
46	1.2E+01	2.7E+01	2.06855E-03	4.28676E-03	6.35531E-03
47	1.2E+01	2.8E+01	9.41719E-04	2.03877E-03	2.98049E-03
48	1.3E+01	2.6E+01	1.87028E-04	3.90763E-04	5.77791E-04
49	1.3E+01	2.7E+01	5.83482E-04	1.27087E-03	1.85435E-03
50	1.3E+01	2.8E+01	6.07720E-04	1.37917E-03	1.98689E-03
51	1.3E+01	2.9E+01	4.74613E-04	1.12162E-03	1.59624E-03
52	1.3E+01	3.0E+01	2.07902E-04	5.11322E-04	7.19225E-04
53	1.4E+01	2.8E+01	4.42883E-04	1.05444E-03	1.49732E-03
54	1.4E+01	2.9E+01	7.52434E-04	1.86125E-03	2.61368E-03
55	1.4E+01	3.0E+01	8.02799E-04	2.06254E-03	2.86534E-03
56	1.4E+01	3.1E+01	6.41279E-04	1.71056E-03	2.35184E-03
57	1.4E+01	3.2E+01	4.09986E-04	1.13494E-03	1.54493E-03
58	1.4E+01	3.3E+01	1.73207E-04	4.97390E-04	6.70597E-04
59	1.5E+01	3.0E+01	6.65518E-05	1.78781E-04	2.45333E-04
60	1.5E+01	3.1E+01	1.43914E-04	4.00666E-04	5.44580E-04
61	1.5E+01	3.2E+01	1.38893E-04	4.00666E-04	5.39559E-04
62	1.5E+01	3.3E+01	1.69787E-04	5.07355E-04	6.77141E-04
63	1.5E+01	3.4E+01	6.66147E-05	2.06139E-04	2.72754E-04
64	1.5E+01	3.5E+01	3.93308E-05	1.25998E-04	1.65329E-04
65	1.6E+01	3.4E+01	1.69621E-04	5.45996E-04	7.15617E-04
66	1.6E+01	3.5E+01	1.70597E-04	5.67811E-04	7.38407E-04
67	1.7E+01	3.7E+01	4.72955E-05	1.74530E-04	2.21825E-04
68	1.8E+01	3.9E+01	4.60356E-05	1.87614E-04	2.33649E-04
69	1.9E+01	4.3E+01	1.73351E-05	8.24288E-05	9.97639E-05
70	2.0E+01	4.7E+01	4.43709E-05	2.44162E-04	2.88533E-04
71	2.1E+01	4.9E+01	3.42360E-05	2.05261E-04	2.39497E-04
72	2.2E+01	5.0E+01	9.23258E-05	5.86315E-04	6.78641E-04
73	2.3E+01	5.2E+01	3.50424E-05	2.41461E-04	2.76503E-04
74	2.4E+01	5.4E+01	6.81551E-05	5.08293E-04	5.76448E-04
75	2.5E+01	5.6E+01	2.60420E-05	2.09705E-04	2.35747E-04

NR.FIT	Z	SIG(MB)	SIGIN	SIGRES	SIGTOT
1	1.0E+00	1.0883E+03	1.21304E+01	2.95521E+02	3.07652E+02
2	2.0E+00	2.3600E+02	1.70424E+01	1.71544E+02	1.88586E+02

3	4.0E+00	2.1000E+00	1.15290E+00	4.81667E-01	1.63457E+00
4	5.0E+00	1.2300E+00	5.01347E-01	2.27437E-01	7.28784E-01
5	6.0E+00	1.0010E+00	1.17386E+00	6.04839E-01	1.77870E+00
6	7.0E+00	2.6000E-01	1.33497E-01	8.43477E-02	2.17844E-01
7	8.0E+00	1.5300E-01	1.11917E-01	9.32811E-02	2.05198E-01
8	9.0E+00	5.0000E-02	1.25996E-02	1.39402E-02	2.65398E-02
9	1.0E+01	4.8000E-02	2.17401E-02	3.06270E-02	5.23671E-02
10	1.1E+01	2.5000E-02	5.37049E-03	9.18278E-03	1.45533E-02
11	1.2E+01	2.5000E-02	9.23244E-03	1.82315E-02	2.74640E-02
12	1.3E+01	2.6000E-02	2.06075E-03	4.67375E-03	6.73449E-03
13	1.4E+01	1.5000E-02	3.22259E-03	8.32111E-03	1.15437E-02
14	1.5E+01	1.3000E-02	6.25091E-04	1.81960E-03	2.44469E-03

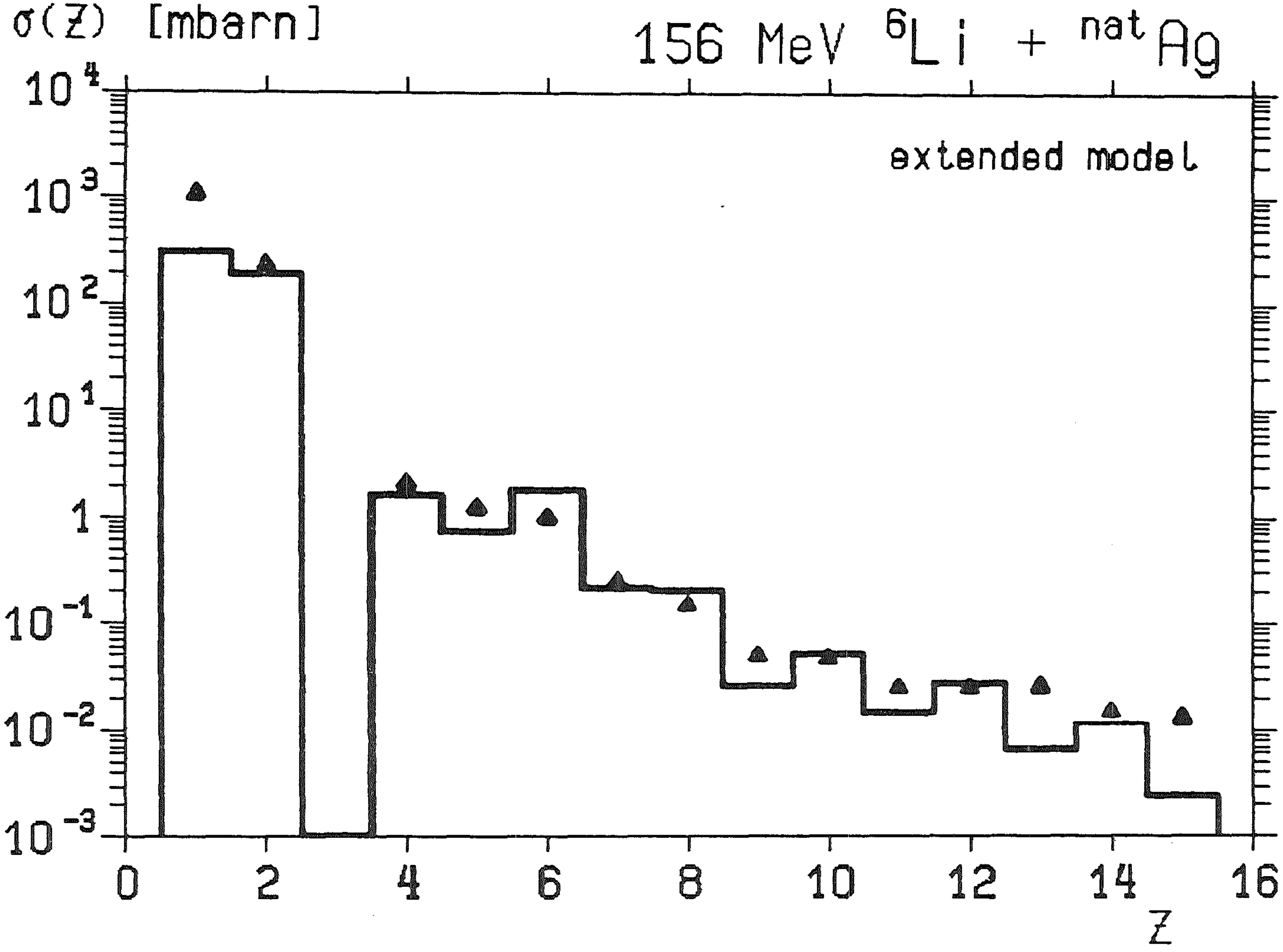
==

VERSION FIF1W1 WITH PROJECTILE-TARGET COMBINATION LIAG

==

OF PROGRAM IMFREM ENDED

=======================


σ(Z) [mbarn]
156 MeV ^{6}Li + ^{nat}Ag
extended model
10^{4}
10^{3}
10^{2}
10^{1}
1
10^{-1}
10^{-2}
10^{-3}
0
2
4
6
8
10
12
14
16
Z

We would like to thank Prof. Dr. H. Rebel for his interest and guiding advice. One of us (I.M.B.) acknowledges the warm hospitality in Kernforschungszentrum Karlsruhe when working out the described program.

4. REFERENCES

1. I.M. Brâncuş, KfK 4453 (to be published 1988)
2. J. Wilczyński, K. Siwek-Wilczyńska, J. van Driel, S. Gonggrijp, D.C.M. Hageman, R.V.F.Janssens, J. Lukasiak and R.M. Siemssen, Phys. Rev. Lett. 45 (1980) 606
3. G.W. Schweimer, unpublished report, Kernforschungszentrum Karlsruhe 1973