THEORETICAL CHEMISTRY INSTITUTE

THE UNIVERSITY OF WISCONSIN

FORTRAN IV PROGRAMS FOR THE EXTRACTION OF POTENTIAL WELL PARAMETERS FROM THE ENERGY DEPENDENCE OF TOTAL ELASTIC SCATTERING CROSS SECTIONS

R. A. LaBudde

TECHNICAL NOTES

PRELIMINARY - NOT FOR PUBLICATION

WIS-TCI-269 17 July 1972

MADISON, WISCONSIN

FORTRAN IV PROGRAMS FOR THE EXTRACTION OF POTENTIAL WELL PARAMETERS FROM THE ENERGY DEPENDENCE OF TOTAL ELASTIC SCATTERING CROSS SECTIONS*

by

R. A. LaBudde

Theoretical Chemistry Institute

University of Wisconsin

Madison, Wisconsin 53706

* This work supported by National Aeronautics and Space Administration Grant NGL 50-002-001 and National Science Foundation Grant GP 26014.

SECTION	SUBSECTION	PAGE	LINE	REMARK
I		3	4	$2\pi/3\ \beta = 2\pi\beta/3$
IV	PHASHT	PL*	2	obscured is C(5)
			3	obscured is - 0.0544
VI	FINAL	PL*	11	obscured is R
VI	SYMQR	PL*	139	Replace 1.D-21 with
				1.D-16 *ENORM/(N*N)
			145	Replace 1.D-21 with
				1.D-16 *ENORM/(N*N)
VII	JWKB	2		Pages 2 and 3 are facing incorrectly
		3		

*PL = Program Listing

PREFACE

An attempt has been made to keep the programs as subroutine oriented as possible. Usually only the main programs are directly concerned with the problem of total cross sections. In particular the subroutines POLFIT (IV.A), BILINR (V.B), GASS59/MAXLIK (VI.D), SYMQR (VI.E), MATIN (VI.F), STUDNT (VI.G), DNTERP (VII.E), DIFTAB (VII.F), FORDIF (VII.G), EPSALG (VII.H), REGFAL (VII.I), AND ADSIMP (VII.J), are completely general, and are concerned only with the problems of numerical analysis and statistics. Each subroutine is independently documented.

TABLE OF CONTENTS

PAGE

I. DISCUSSION*

The total cross section $Q(v)$, where v is the relative velocity, is assumed to be a sum of two components: $Q_{SLL}(v)$, dependent upon the long-range force constant C_6, and $\Delta Q_g(v)$, an oscillatory term due to the presence of glory extrema

$$Q(v) = Q_{SLL}(v) + \Delta Q_g(v) \tag{I-1}$$

where

$$Q_{SLL}(v) \simeq 8.083 \left(\frac{C_6}{\hbar v}\right)^{2/5} \tag{I-2a}$$

$$\Delta Q_g(v) \simeq \frac{4\pi^{3/2}}{k^2} \frac{L_g}{(-\eta_m'')^{1/2}} \sin\left(2\eta_m - \frac{3\pi}{4}\right) \tag{I-2b}$$

where $k = \mu v/\hbar$ is the wavenumber, L_g is value of the orbital quantum number ℓ corresponding to η_m, a maximum in the phase shift, and η_m'' is the second-derivative of η_m with respect to ℓ evaluated at L_g.

It can be shown that by expanding about the high-velocity limit, (2b) may be written

$$\Delta Q_g(v) = (2\pi r_m)^{3/2} g_0 \left(\frac{\hbar v}{\epsilon}\right)^{1/2} \left[1 + \frac{H_1\epsilon}{E} + \frac{H_2\epsilon^2}{E^2} + \cdots\right] \sin\phi \tag{I-3}$$

where $E = \frac{1}{2}\mu v^2$ is the energy,

$$\phi = -\frac{3\pi}{4} + \frac{4\epsilon r_m}{\hbar v}\left(a_1 + \frac{A_1\epsilon}{E} + \frac{A_2\epsilon^2}{E^2} + \cdots\right) \tag{I-4}$$

---

* For a more complete description of the derivations of the formulae see Wisc. Theor. Chem. Institute Report WIS-TCI-469 (1972).

---

and r_m is the position of the minimum of (of depth ϵ) the potential. The constants g_o, a_1 are related to the curvature of the potential well at $r = r_m$, and the constants H_1, H_2, ... , A_1, A_2, ... , were found to be to a good approximation independent of the curvature, ϵ, and r_m, and of the form of the potential. Furthermore, for the energy range $1 \leq E^* \leq 100$, where $E^* = E/\epsilon$ is the reduced energy, it was found only the terms H_1 and A_1, A_2, A_3 and A_4 were required. For a Lennard-Jones 12-6 potential, i.e.,

$$V(r) = \epsilon \left[\left(\frac{r_m}{r} \right)^{12} - 2 \left(\frac{r_m}{r} \right)^{6} \right]$$

the values of the constants are:

$$g_o = 0.186299 \tag{I-5a}$$

$$H_1 = 3.267 \pm 0.80 \tag{I-5b}$$

$$a_1 = 0.4215587 \tag{I-5c}$$

$$A_1 = -0.1655 \pm .0016 \tag{I-5d}$$

$$A_2 = 0.1057 \pm 0.0160 \tag{I-5e}$$

$$A_3 = -0.0544 \pm 0.0160 \tag{I-5f}$$

$$A_4 = 0.0139 \pm 0.0084 \tag{I-5g}$$

In terms of free parameters, (1) may be rewritten

$$Q(v) = \gamma_1 v^{-2/5} + \gamma_2 v^{1/2} \left(1 + \frac{H_1 \epsilon}{E}\right) \sin\phi \tag{I-6}$$

and

$$\phi = -\frac{3\pi}{4} + \frac{\gamma_3}{v} + \frac{\gamma_4}{vE} \left(A_1 + \frac{A_2 \epsilon}{E} + \frac{A_3 \epsilon^2}{E^2} + \frac{A_4 \epsilon^3}{E^3} \right) \tag{I-7}$$

where

$$\gamma_1 = 8.083 \left(\frac{C_6}{\hbar}\right)^{2/5}$$

$$\gamma_2 = (2\pi r_m)^{3/2} g_o (\hbar/\epsilon)^{1/2}$$

$$\gamma_3 = 4 a_1 \epsilon r_m / \hbar$$

$$\gamma_4 = 4 \epsilon^2 r_m / \hbar \qquad \text{(I-8)}$$

It has been shown previously that when the r^{-8} term in the potential begins to affect the cross section, it may be accounted for by replacing $Q_{SLL}(v)$ in (1) by $Q_{SLL}(v) + 2\pi/3\ \beta$, where $\beta = C_8/C_6$.

It is clear from the form of (I-6) that $Q(v)$ has an oscillatory component. The condition for extrema in Q is that $\sin\phi = \pm 1$, which can be shown to lead to:

$$\left(N - \frac{3}{8}\right) v_g = \frac{2\epsilon r_m}{\hbar}\left(a_1 + \frac{A_1\epsilon}{E} + \frac{A_2\epsilon^2}{E^2} + \frac{A_3\epsilon^3}{E^3} + \frac{A_4\epsilon^4}{E^4}\right) \qquad \text{(I-9)}$$

where the maxima and minima in $Q(v)$ are indexed by $N(v_g)$ in the order they occur, i.e., N = 1 corresponds to the first maximum, N = 1.5 corresponds to the first minimum, N = 2 is the second maximum, etc. The quantities v_g are the associated values of v at the extrema.

II. UNITS

All of the results and programs below assume the following system of units:

Mass	$M = 10^{-24}$ g	= 1 ppg (picopicogram)
Length	$L = 10^{-8}$ cm	= 1 Å (Ångström)
Time	$T = 10^{-13}$ sec	= 1 dps (decipicosecond)

In this sytem,

1 amu = 1.660531 ppg

1 a.u. = 0.52917715 Å

1 Km/sec = 1 Å/dps

1 eV = 160.21917 cpe

1 Kcal/mole = 6.94793 cpe

h = 6.626196 cpe-dps

$\hbar$ = 1.0545919 cpe-dps

kT = 1.380622 cpe @ T = 100°K

III. XSECT

The purpose of the main program XSECT is to calculate total cross sections $Q(v)$ by means of (I—6)-(I—8). The assumption is made that H_1 and A_1, A_2, A_3, A_4 are those for the L-J(12-6)potential given in (I—5).

The first card of the data set contains (18A4) 72 columns of title information, which is reproduced at the top of the output.

The second card of the data specifies N, MU, EPS, A1, BO, C1, C6, in that order, according to the format (I5, 6F10.4, E15.6), where:

N - No. of velocities for which $Q(v)$ is to be calculated

MU - μ (reduced mass)

EPS - ε

RM - r_m

A1 - a_1 (I-5c)

BO, C1 - b_o and c_1 are related to the leading terms in the expansion in powers of $1/E^*$ of L_g and η_m'' respectively. Since the only quantity required is $g_o = b_o/c_1^{1/2}$, b_o may be set equal to g_o and c_1 to 1.

C6 = C_6 the long-range force constant.

The succeeding cards of the data contain the velocities V(I), I = 1, ..., N, for which the Q(v) are to be calculated, according to the format (8F10.4).

The output of the program is a table of the values of V(I), QSLL, DELQ, Q, and QF for I = 1, ..., N, where

$$QSLL = Q_{SLL}(v)$$

$$DELQ = \Delta Q_g(v)$$

$$Q = Q(v) = Q_{SLL}(v) + \Delta Q_g(v)$$

$$QF = \Delta Q_g(v)/Q_{SLL}(v)$$

```
XSECT
O COMPILED BY FOR-V S7E6 ON 06 SEP 71 AT 17:41:11.

GE USED: CODE(1) 000270; DATA(0) 000515

  1*    C       PROGRAM TO GENERATE RAPID CROSS SECTIONS
  2*            REAL MU
  3*            DIMENSION V(200),A(4),TITLE(18)
  4*            DATA A/-0.1655,0.1057,-0.0544,0.0139/,H1/3.267/
  5*            DATA PI/3.14159265/,HCROSS/1.0545919/
  6*    100     READ (5,5) (TITLE(I),I=1,18)
  7*    5       FORMAT (18A4)
  8*            WRITE (6,6) (TITLE(I),I=1,18)
  9*    6       FORMAT (1H1,10X,18A4)
 10*            READ (5,10) N,MU,EPS,RM,A1,B0,C1,C6
 11*    10      FORMAT (I5,6F10.4,E15.6)
 12*            WRITE (6,20) N,MU,EPS,RM,A1,B0,C1,C6
 13*    20      FORMAT (1H0,10X,'NO. OF VELOCITIES',I8/11X,'REDUCED MASS',E15.6/
 14*           1 11X,'WELL-DEPTH EPS',E15.6/11X,'POTENTIAL MINIMUM RM',E15.6/11X,
 15*           2 'PHASE SHIFT COEFFICIENT A1',E15.6/11X,'BETA COEFFICIENT B0',
 16*           3 E15.6/11X,'COEFFICIENT C1',E15.6/11X,'LONG RANGE C6',E15.6/)
 17*            READ (5,30) (V(I),I=1,N)
 18*    30      FORMAT (8F10.4)
 19*            D1 = C6/HCROSS
 20*            D2 = 2.*PI*RM*B0
 21*            D3 = 4.*RM/HCROSS
 22*            D4 = 2.*PI*RM*HCROSS/(EPS*C1)
 23*            WRITE (6,40)
 24*    40      FORMAT (//1H0,5X,'NO.', 8X,'V',13X,'QSLL',10X,'DELQ',11X,'QTOT',
 25*           1 10X,'DELQ/QSLL'/4X,100(1H*)/)
 26*            DO 200 I = 1,N
 27*            QSLL = 8.083*(D1/V(I))**0.4
 28*            ETR = 0.5*MU*V(I)*V(I)
 29*            Z = EPS/ETR
 30*            PHI = - 0.75*PI + D3*EPS*(A1 + Z*(A(1) + Z*(A(2) + Z*(A(3) + Z*
 31*           1 A(4))))/V(I)
 32*            DELQ = D2*SQRT(D4*V(I))*(1. + H1*Z)*SIN(PHI)
 33*            Q = QSLL + DELQ
 34*            QF = DELQ/QSLL
 35*    200     WRITE (6,50) I,V(I),QSLL,DELQ,Q,QF
 36*    50      FORMAT (3X,I5,5E15.6)
 37*            GO TO 100
 38*            END

ND OF COMPILATION:        NO  DIAGNOSTICS.


      COMPILATION TIME:          0.653 SEC.
      I/O REQUESTS:                50
      I/O WORDS TRANSFERED:      57470
```

IV. PHASHT

The purpose of this program is to produce a fit of the experimental variable $Y = (N - \frac{3}{8})v_N$ to the model

$$Y = I + \frac{S_1}{E} + \frac{S_2}{E^2} + \cdots \qquad \text{(IV-1)}$$

where, after regression, the following ansatz is made:

$$I = \frac{2a_1 \epsilon r_m}{\pi\hbar} \quad ; \quad S_i = \frac{2\epsilon^{i+1} r_m}{\pi\hbar} A_i \,, \quad i=1,2,\cdots \qquad \text{(IV-2)}$$

which leads to

$$\begin{aligned} a_1 \epsilon r_m &= \frac{\pi\hbar}{2} I \\ \epsilon^2 r_m &= \frac{\pi\hbar}{2} S_1/A_1 \\ \epsilon &= \frac{A_1 S_2}{A_2 S_1} \end{aligned} \qquad \text{(IV-3)}$$

Values of $\epsilon^2 r_m$ and ϵ are calculated from (IV-2) assuming A_1, A_2, A_3, A_4 are given by (I-5). "ERROR" denotes 95% confidence interval halfwidths. The confidence intervals were calculated using first-order propagation of error,** and took the error in A_1, A_2, A_3, A_4 into account.

** I.e., if $y = f(x_1, x_2)$, then

$$\sigma_y^2 \simeq \left(\frac{\partial f}{\partial x_1}\right)^2 \sigma_{x_1}^2 + 2\left(\frac{\partial f}{\partial x_1}\right)\left(\frac{\partial f}{\partial x_2}\right) \mathrm{cov}(x_1, x_2) + \left(\frac{\partial f}{\partial x_2}\right)^2 \sigma_{x_2}^2$$

Input to the program is as follows:

Card 1: title information in columns 1-72 (18A4)

Card 2: NOB, IWGHT, MU (2I5, F10.4)

Card 3 (or more): V(I), I = 1, ..., NOB (8F10.4)

Card 4 (or more): INDEX(I), I = 1, ..., NOB (8F10.4)

where

NOB - number of observed extrema

IWGHT - if 0 data has assumed constant absolute error. If 1 , data is assumed to have constant relative error.

MU = μ (reduced mass)

V(I) - extrema velocity v_N for I-th extremum

INDEX(I) - extrema index N for I-th extremum

```
R,SI PHASHT
LE 000 COMPILED BY FOR~V S7E6 ON 04 JAN 72 AT 08:33:58.

STORAGE USED: CODE(1) 000424: DATA(0) 000562

01     1.          REAL INDEX (30),MU
03     2.          DIMENSION X(30),Y(30),W(30),V(30),A(4),C(8),E(8),R(5,5),TITLE(10)
04     3.          DATA A1/0.421559/,A/-0.1655,0.1057,-0.0554,0.0137/,CONS/1.65547E
04     4.         1/,ERA1/0.9895E-4/,ERA1A2/.6193E-2/
09     5.    C     READ IN EXPERIMENT TITLE
12     6.    100   READ (5,5) (TITLE(I),I=1,10)
20     7.    5     FORMAT (10A4)
21     8.          WRITE (6,6) (TITLE(I),I=1,10)
27     9.    6     FORMAT (1H1,10X,10A4/)
27    10.    C     READ IN NO. OF DATA POINTS NOB, WEIGHT SWITCH IWGHT (IF 0 ABSOLUT
27    11.    C      ERROR AND W(I) = 1.  IF 1, RELATIVE ERROR AND W(I) = 1/Y(I)**2),
27    12.    C     AND REDUCED MASS MU.
27    13.    C     NOTE ... IF V IN UNITS OF KM/SEC AND MU IN 1.E-24 GRAM, THEN UNIT
27    14.    C     OF ENERGY WILL BE 1.E-14 ERGS, (NOTE THAT CONS HAS UNITS OF ACTIO
27    15.    C      1.E-27 ERG-SEC)
30    16.          READ (5,10) NOB,IWGHT,MU
35    17.    10    FORMAT (2I5,F10.4)
36    18.          WRITE (6,20) NOB,IWGHT,MU
43    19.    20    FORMAT (1HO,10X,'NO. OF OBSERVATIONS ...',I10/11X,'WEIGHTING SWIT
43    20.         1H ...',I10/11X,'REDUCED MASS ...',E15.6)
43    21.    C     READ IN EXTREMA VELOCITIES AND EXTREMA INDICES
44    22.          READ (5,30) (V(I),I=1,NOB)
52    23.          READ (5,30) (INDEX(I),I=1,NOB)
60    24.    30    FORMAT (8F10.4)
61    25.          WRITE (6,40)
63    26.    40    FORMAT (////5X,'NO.', 8X,'V',18X,'ETR',11X,'INDEX',11X,'X',14X,'Y
63    27.         1/6X,80(1H*)/)
64    28.          DO 500 I = 1,NOB
67    29.          ETR = 0.5*MU*V(I)*V(I)
70    30.          X(I) = 1./ETR
71    31.          Y(I) = (INDEX(I) - 0.375)*V(I)
72    32.          W(I) = 1.
73    33.          IF (IWGHT .EQ. 1) W(I) = 1./(Y(I)*Y(I))
75    34.    500   WRITE (6,50) I,V(I),ETR,INDEX(I),X(I),Y(I)
06    35.    50    FORMAT (3X,I5,5E15.6)
07    36.          CALL POLFIT (X,Y,W,NOB,C,E,R,NDEG,SIG)
10    37.          WRITE (6,60) NDEG,SIG
14    38.    60    FORMAT (//10X,'POLYNOMIAL OF DEGREE',I5,' DETERMINED'/10X,'RMS R
14    39.         1SIDUAL ...',E15.5//11X,'TERM',8X,'COEFFICIENT',10X,'ERROR'/5X,50
14    40.         2(1H*)/)
15    41.          NN = NDEG + 1
16    42.          DO 600 I = 1,NN
21    43.          J = I-1
22    44.    600   WRITE (6,70) J,C(I),E(I)
30    45.    70    FORMAT (9X,I5,E15.5,5X,E15.5)
31    46.          AER = CONS*C(1)
32    47.          EE = CONS*E(1)
33    48.          EPSRM = AER/A1
34    49.          EE2= EE/A1
35    50.          WRITE (6,80) AER,EE,EPSRM,EE2
43    51.    80    FORMAT (/10X,'A1*EPS*RM =',E12.5,2X,'(ERROR=',E12.5,' )'/10X,'EPS
43    52.         1RM FROM A1 = 0.421559 ...',E12.5,2X,'(ERROR =',E12.5,' )')
```

```
244     53*           IF (NDEG .EQ. 0) GO TO 100
246     54*           EER = CONS*C(2)/A(1)
247     55*           EE = ABS(CONS*E(2)/A(1))
250     56*           EPS = EER/EPSRM
251     57*           EE2 = EPS*SQRT(R(1,1)/C(1)**2 + R(2,2)/C(2)**2 -2.*R(1,2)/(C(1)*C
251     58*          12)) + ERA(1)
252     59*           WRITE (6,85) EER,EE,EPS,EE2
260     60*    85     FORMAT(/10X,'EPS*EPS*RM =',E12.5,2X,'(ERROR =',E12.5,' )'/10X,'EP
260     61*          1 FROM A1 =0.421559 ...',E12.5,2X,'(ERROR =',E12.5,' )')
261     62*           IF (NDEG .EQ. 1) GO TO 100
263     63*           EPS = C(3)*A(1)/(C(2)*A(2))
264     64*           EE = EPS*SQRT(R(2,2)/C(2)**2 + R(3,3)/C(3)**2 - 2.*R(2,3)/(C(2)*
264     65*          1 C(3)) + ERA(A2)
265     66*           WRITE (6,90) EPS,EE
271     67*    90     FORMAT (/10X,'EPS =',E12.5,2X,,'(ERROR =',E15.5,' )')
272     68*           GO TO 100
273     69*           END
```

END OF COMPILATION: NO DIAGNOSTICS.

COMPILATION TIME: 0.959 SEC.
I/O REQUESTS: 61
I/O WORDS TRANSFERED: 61198

A. POLFIT

This subroutine fits a polynomial

$$p(x) = \sum_{k=0}^{M} c_k x^k$$

to a set of N data points (x_i, y_i), $1 \le i \le N$, by weighted forward regression. The criterion used to determine $c_0, c_1, \ldots, c_M$ is the minimization of the approximate variance of fit s^2:

$$s^2 = \frac{1}{N-M-1} \sum_{i=1}^{N} w_i \left(y_i - p(x_i) \right)^2$$

where w_i is the weight assigned to the i-th data point (if the variance of the i-th data point is σ_i^2, then $w_i \propto 1/\sigma_i^2$).

The mechanics of the fit are performed by use of calculated orthogonal polynomials:

$$p(x) = \sum_{k=0}^{M} a_k P_k(x)$$

where the P_k's satisfy $P_0(x) = 1$ and

$$P_{k+1}(x) = (x - u_k) P_k(x) - v_k P_{k-1}(x)$$

$$\sum_{i=1}^{N} w_i P_k(x_i) P_j(x_i) = \delta_{kj} D_k$$

The fit is continued until all terms whose contributions to the variance s^2 are significant at the 90% level (F-test) have been added.

CALL POLFIT (X, Y, W, N, C, E, R, M, SIG)

Input:

X(I) - abscissa x_i of the I-th data point

Y(I) - ordinate y_i of the I-th data point

W(I) - weight w_i of the I-th data point

N - no. of data points

Output:

C(I) - c_{i-1} , coefficient of x^{i-1} in final polynomial

E(I) - 95% confidence interval halfwidth of C(I), i.e., $t\,\sigma_I$

R(I,J) - covariance of C(I) and C(J) multiplied by t^2 ,

i.e., t^2 cov (I,J)

M - degree of final fitted polynomial

SIG - S , standard error of fit where t is the critical value of the student-t statistic for $\alpha = .05$ (Equal-Tails) and N-M-1 degrees of freedom.

NOTE: Because of dimensioning, the program in its given form is limited to $N \leq 21$ and $M \leq 4$.

```
OR,SI POLFIT
CLE 000 COMPILED BY FOR-V S7E6 ON 04 JAN 72 AT 08:34:03.

 STORAGE USED: CODE(1) 000662: DATA(0) 000323

101      1*            SUBROUTINE POLFIT (X,Y,W,N,C,E,R,M,SIG)
101      2*      C     FITS A POLYNOMIAL OF DEGREE M (DETERMINED BY ROUTINE) TO A SET OF
101      3*      C     N POINTS (X(I),Y(I)) I=1,...,N .
101      4*      C     W(I) IS THE WEIGHT ASSIGNED TO POINT I. E.G. FOR ASSUMED CONSTANT
101      5*      C     ABSOLUTE ERROR, W(I) = 1. , AND FOR ASSUMED CONSTANT PERCENTAGE
101      6*      C     ERROR, W(I) = 1./(Y(I)*Y(I)).
101      7*      C     THE POLYNOMIAL IS GIVEN BY P(X) =  C(1) + ... C(M+1)*X**M
101      8*      C     E(I) IS THE 95 PERCENT CONFIDENCE INTERVAL HALFWIDTH OF C(I)
101      9*      C     R(I,J) IS THE COVARIANCE BETWEEN C(I) AND C(J), MULTIPLIED BY THE
101     10*      C     SQUARE OF THE CRITICAL VALUE OF THE STUDENT T STATISTIC
101     11*      C     THE DEGREE IS INCREASED UNTIL ALL TERMS SIGNIFICANT AT THE 90
101     12*      C     PERCENT LEVEL HAVE BEEN INCLUDED.
101     13*      C     THE POLYNOMIAL IS DETERMINED USING THE METHOD OF ORTHOGONAL
101     14*      C     POLYNOMIALS.
101     15*      C     AT EXIT FROM THE ROUTINE M WILL BE THE DETERMINED DEGREE OF THE
101     16*      C     POLYNOMIAL AND SIG WILL CONTAIN THE RMS RESIDUAL OF THE FIT
103     17*            DOUBLE PRECISION U,V,A,B,VY,AYY,WAYY,SYN,T,WORK,P1,P2,SUM1,SUM2,
103     18*           1 SUM3,SUM4
104     19*            DIMENSION X(1),Y(1),W(1),C(1),E(1),U(5),V(5),A(5),B(5,5),D(5),
104     20*           1 R(5,5),F90(25),T95(25)
105     21*            EQUIVALENCE (VY,B(1,5)),(T,B(1,4)),(WAYY,B(1,3)),(SYN,B(2,5)),
105     22*           1 (WORK,B(2,4)),(AYY,B(3,5))
106     23*            DATA T95/12.706,4.303,3.182,2.776,2.571,2.447,2.365,2.306,2.262,
106     24*           1 2.228,2.201,2.179,2.160,2.145,2.131,2.120,2.110,2.101,2.093,2.08
106     25*           2, 2.080,2.074,2.069,2.064,2.060/
110     26*            DATA F90/39.86,8.53,5.54,4.54,4.06,3.78,3.59,3.46,3.36,3.28,3.23,
110     27*           1 3.18,3.14,3.10,3.07,3.05,3.03,3.01,2.99,2.97,2.96,2.95,2.94,2.93
110     28*           2 2.92/
112     29*            DATA V(1)/0.D0/
114     30*            DO 200 I = 1,4
117     31*            II = I + I
120     32*            DO 100 J = 1,II
123     33*      100   B(I,J) = 0.D0
125     34*            B(5,I) = 0.D0
126     35*      200   B(I,I) = 1.D0
130     36*            B(5,5) = 1.D0
131     37*            SUM1 = 0.D0
132     38*            SUM2 = 0.D0
133     39*            SUM3 = 0.D0
134     40*            VY = 0.D0
135     41*            DO 300 I = 1,N
140     42*            T = W(I)
141     43*            SUM1 = SUM1 + T
142     44*            SUM2 = SUM2 + T*X(I)
143     45*            T = T*Y(I)
144     46*            SUM3 = SUM3 + T
145     47*      300   VY = VY + T*Y(I)
147     48*            A(1) = SUM3/SUM1
150     49*            D(1) = SUM1
151     50*            U(1) = SUM2/SUM1
152     51*            B(2,1) = - U(1)
153     52*            AYY = 0.D0
```

```
154     53*             VY = VY - A(I)*A(I)*D(I)
155     54*             I = I
156     55*     500     I = I + 1
157     56*             IF (I .GT. 5) GO TO 1100
161     57*             II = I - 1
162     58*             SUM1 = 0.D0
163     59*             SUM2 = 0.D0
164     60*             SUM3 = 0.D0
165     61*             SUM4 = 0.D0
166     62*             DO 800 J = 1,N
171     63*             P1 = 1.D0
172     64*             P2 = 0.D0
173     65*             WORK = X(J)
174     66*             DO 700 K = 1,II
177     67*             T = (WORK -U(K))*P1 - V(K)*P2
200     68*             P2 = P1
201     69*     700     P1 = T
203     70*             T = W(J)*P1
204     71*             SUM1 = SUM1 + T*P1*WORK
205     72*             SUM2 = SUM2 + T*P2*WORK
206     73*             SUM3 = SUM3 + T*P1
207     74*     800     SUM4 = SUM4 + T*Y(J)
211     75*             D(I) = SUM3
212     76*             U(I) = SUM1/SUM3
213     77*             V(I) = SUM2/D(II)
214     78*             A(I) = SUM4/SUM3
215     79*             WAYY = A(I)*A(I) *D(I)
216     80*             AYY = AYY + WAYY
217     81*             IF (I.EQ.2) GO TO 1000
221     82*             B(I,1) = -U(II)*B(II,1) - V(II)*B(I-2,1)
222     83*             DO 900 J = 2,II
225     84*     900     B(I,J) = B(II,J-1) - U(II)*B(II,J) - V(II)*B(I-2,J)
227     85*     1000    IDF = N-I
230     86*             SYN = DABS(VY - AYY)/IDF
231     87*             FP = WAYY/SYN
232     88*             IF (FP .GT. F90(IDF)) GO TO 500
234     89*     1100    MM = I-1
235     90*             M = MM-1
236     91*             AYY = AYY - WAYY
237     92*             IDF = N-MM
240     93*             SYN = DABS(VY - AYY)/IDF
241     94*             DO 1600 K = 1,MM
244     95*             T = 0.D0
245     96*             DO 1500 J = K,MM
250     97*     1500    T = T + A(J)*B(J,K)
252     98*     1600    C(K) = T
254     99*             COEF = SYN*T95(IDF)**2
255    100*             DO 2000 K = 1,MM
260    101*             DO 1900 J = K,MM
263    102*             T = 0.D0
264    103*             DO 1800 L = J,MM
267    104*     1800    T = T + B(L,K)*B(L,J)/D(L)
271    105*             R(K,J) = T*COEF
272    106*     1900    R(J,K) = R(K,J)
274    107*     2000    E(K) = SQRT(R(K,K))
276    108*             SIG = SQRT(SYN)
277    109*             RETURN
```

```
300    110.          END

     END OF COMPILATION:        NO  DIAGNOSTICS.

                COMPILATION TIME:       1.526 SEC.
                I/O REQUESTS:             66
                I/O WORDS TRANSFERED:   62865
```

B. EXAMPLES

Four example calculations are given.

1. Data of U. Buck, K. A. Köhler, H. Pauly, Z. Phys. 244, 180 (1971). IWGHT = 1 (assumed constant relative error), so the rms residual S for the fit is an estimate of the coefficient of variation of the data. The value S = 0.31% is compatible with known experimental precision.

Examples 2-4 are based upon the synthetic data calculated from (IV-1) with I, S_1, S_2, S_3, S_4, given by (IV-2), with ε = 8 cpe, r_m = 4 Å, a_1 = 0.421559. I.e.,

$$\hat{Y}_i = I + \frac{S_1}{E_i} + \frac{S_2}{E_i^2} + \frac{S_3}{E_i^3} + \frac{S_4}{E_i^4} \quad (1 \le i \le 13)$$

and $Y_i = \hat{Y}_i (1+\delta)$ where δ is a normal distributed random variable with mean 0 and variance σ^2. In each case IWGHT = 1, so the calculated value of S should approximate σ.

2. $Y = \hat{Y}$, i.e., $\sigma = 0$. Calculated S = 0.0098%.

3. σ = 0.5%. Calculated S = .55%.

4. σ = 0.25%. Calculated S = 0.18%.

DATA OF U. BUCK, K.A. KOHLER, AND H. PAULY , Z. PHYSIK 244 180 (1971)

NO. OF OBSERVATIONS ... 13
WEIGHTING SWITCH ... 1
REDUCED MASS ... .342499+02

NO.	V	ETR	INDEX	X	Y
1	.234000+01	.937694+02	.500000+01	.106645-01	.108225+02
2	.207500+01	.737336+02	.550000+01	.135623-01	.106344+02
3	.188000+01	.605264+02	.600000+01	.165217-01	.105750+02
4	.170500+01	.497827+02	.650000+01	.200813-01	.104431+02
5	.154500+01	.408777+02	.700000+01	.244632-01	.102356+02
6	.141300+01	.341911+02	.750000+01	.292473-01	.100676+02
7	.130500+01	.291642+02	.800000+01	.342886-01	.995062+01
8	.120700+01	.249485+02	.850000+01	.400826-01	.980687+01
9	.112500+01	.216738+02	.900000+01	.461387-01	.970312+01
10	.104500+01	.187009+02	.950000+01	.534734-01	.953562+01
11	.968000+00	.160465+02	.100000+02	.623189-01	.931700+01
12	.910000+00	.141812+02	.105000+02	.705160-01	.921375+01
13	.855000+00	.125188+02	.110000+02	.798801-01	.908437+01

POLYNOMIAL OF DEGREE 3 DETERMINED
RMS RESIDUAL 31262-02

TERM	COEFFICIENT	ERROR
0	.11338+02	.15915+00
1	-.55601+02	.14017+02
2	.52421+03	.34821+03
3	-.22900+04	.25523+04

A1*EPS*RM = .18782+02 (ERROR= .26364+00)
EPS*RM FROM A1 = 0.421559 ... .44553+02 (ERROR = .62538+00)

EPS*EPS*RM = .55653+03 (ERROR = .14030+03)
EPS FROM A1 =0.421559 ... .12492+02 (ERROR = .29827+01)

EPS = .14762+02 (ERROR = .62868+01)

TEST DATA --- EPS = 8. --- RM = 4. --- NO ERROR

NO. OF OBSERVATIONS ... 13
WEIGHTING SWITCH ... 1
REDUCED MASS ... .300000+02

NO.	V	ETR	INDEX	X	Y
1	.800000+00	.960000+01	.839090+01	.104167+00	.641272+01
2	.900000+00	.121500+02	.778734+01	.823045-01	.667111+01
3	.100000+01	.150000+02	.725479+01	.666667-01	.687979+01
4	.110000+01	.181500+02	.678422+01	.550964-01	.705014+01
5	.120000+01	.216000+02	.636708+01	.462963-01	.719049+01
6	.130000+01	.253500+02	.599584+01	.394477-01	.730710+01
7	.140000+01	.294000+02	.566407+01	.340136-01	.740470+01
8	.150000+01	.337500+02	.536633+01	.296296-01	.748699+01
9	.160000+01	.384000+02	.509803+01	.260417-01	.755685+01
10	.170000+01	.433500+02	.485533+01	.230681-01	.761656+01
11	.180000+01	.486000+02	.463495+01	.205761-01	.766790+01
12	.190000+01	.541500+02	.443412+01	.184672-01	.771232+01
13	.200000+01	.600000+02	.425049+01	.166667-01	.775097+01

POLYNOMIAL OF DEGREE 4 DETERMINED
RMS RESIDUAL 97530-04

TERM	COEFFICIENT	ERROR
0	.81434+01	.10321-01
1	-.25576+02	.96037+00
2	.13067+03	.29574+02
3	-.54394+03	.36361+03
4	.11076+04	.15322+04

A1*EPS*RM = .13490+02 (ERROR= .17097-01)
EPS*RM FROM A1 = 0.421559 ... .32000+02 (ERROR = .40556-01)

EPS*EPS*RM = .25600+03 (ERROR = .96126+01)
EPS FROM A1 =0.421559 ... .80000+01 (ERROR = .30107+00)

EPS = .79998+01 (ERROR = .16387+01)

TEST DATA --- EPS = 8. --- RM = 4. --- SIGMA = 0.005

NO. OF OBSERVATIONS ... 13
WEIGHTING SWITCH ... 1
REDUCED MASS ... .300000+02

NO.	V	ETR	INDEX	X	Y
1	.800000+00	.960000+01	.826601+01	.104167+00	.631281+01
2	.900000+00	.121500+02	.780906+01	.823045-01	.669065+01
3	.100000+01	.150000+02	.721831+01	.666667-01	.684331+01
4	.110000+01	.181500+02	.674615+01	.550964-01	.700827+01
5	.120000+01	.216000+02	.628429+01	.462963-01	.709115+01
6	.130000+01	.253500+02	.598548+01	.394477-01	.729362+01
7	.140000+01	.294000+02	.568245+01	.340136-01	.743044+01
8	.150000+01	.337500+02	.531429+01	.296296-01	.740894+01
9	.160000+01	.384000+02	.511698+01	.260417-01	.758717+01
10	.170000+01	.433500+02	.487927+01	.230681-01	.765726+01
11	.180000+01	.486000+02	.464704+01	.205761-01	.768967+01
12	.190000+01	.541500+02	.446258+01	.184672-01	.776641+01
13	.200000+01	.600000+02	.426992+01	.166667-01	.778985+01

POLYNOMIAL OF DEGREE 3 DETERMINED
RMS RESIDUAL 54587-02

TERM	COEFFICIENT	ERROR
0	.84388+01	.23523+00
1	-.44189+02	.15733+02
2	.44756+03	.29728+03
3	-.20993+04	.16483+04

A1*EPS*RM = .13979+02 (ERROR= .38967+00)
EPS*RM FROM A1 = 0.421559 ... .33161+02 (ERROR = .92436+00)

EPS*EPS*RM = .44231+03 (ERROR = .15748+03)
EPS FROM A1 =0.421559 ... .13338+02 (ERROR = .43880+01)

EPS = .15858+02 (ERROR = .51957+01)

TEST DATA --- EPS = 8. --- RM = 4. --- SIGMA = 0.0025

NO. OF OBSERVATIONS ... 13
WEIGHTING SWITCH ... 1
REDUCED MASS ... .300000+02

NO.	V	ETR	INDEX	X	Y
1	.800000+00	.960000+01	.838536+01	.104167+00	.640828+01
2	.900000+00	.121500+02	.780310+01	.823045-01	.668529+01
3	.100000+01	.150000+02	.727100+01	.666667-01	.689600+01
4	.110000+01	.181500+02	.680110+01	.550964-01	.706871+01
5	.120000+01	.216000+02	.636197+01	.462963-01	.718436+01
6	.130000+01	.253500+02	.598382+01	.394477-01	.729147+01
7	.140000+01	.294000+02	.566941+01	.340136-01	.741217+01
8	.150000+01	.337500+02	.535622+01	.296296-01	.747183+01
9	.160000+01	.384000+02	.511618+01	.260417-01	.758589+01
10	.170000+01	.433500+02	.485833+01	.230681-01	.762166+01
11	.180000+01	.486000+02	.464257+01	.205761-01	.768163+01
12	.190000+01	.541500+02	.443228+01	.184672-01	.770883+01
13	.200000+01	.600000+02	.425656+01	.166667-01	.776312+01

POLYNOMIAL OF DEGREE 3 DETERMINED
RMS RESIDUAL 18440-02

TERM	COEFFICIENT	ERROR
0	.81921+01	.79429-01
1	-.28619+02	.53186+01
2	.18464+03	.10056+03
3	-.71384+03	.55802+03

A1*EPS*RM = .13571+02 (ERROR= .13158+00)
EPS*RM FROM A1 = 0.421559 ... .32192+02 (ERROR = .31212+00)

EPS*EPS*RM = .28646+03 (ERROR = .53236+02)
EPS FROM A1 =0.421559 ... .88986+01 (ERROR = .15719+01)

EPS = .10101+02 (ERROR = .37469+01)

V. XTREMA

This main program produces a fit of the experimental variable $Y = (N - \frac{3}{8})v_N$ to the model

$$Y = c_1 + \frac{c_2}{E}\left(A_1 + \frac{A_2 c_3}{E} + \frac{A_3 c_3^2}{E^2} + \frac{A_4 c_3^3}{E^3}\right) \qquad \text{(V-1)}$$

where, after regression the following ansatz is made:

$$c_1 = 2a_1 \epsilon r_m / \pi t_h \; ; \; c_2 = 2\epsilon^2 r_m / \pi t_h \; ; \; c_3 = \epsilon \qquad \text{(V-2)}$$

which lead to

$$\begin{aligned} a_1 \epsilon r_m &= \frac{\pi t_h}{2} c_1 \\ \epsilon^2 r_m &= \frac{\pi t_h}{2} c_2 \\ \epsilon &= c_3 \end{aligned} \qquad \text{(V-3)}$$

The details of the fitting procedure are described in V.A (FIT).

The description of the input and output is the same as in Section IV.

```
SI XTREMA
 000 COMPILED BY FOR-V S7E6 ON 16 SEP 71 AT 19:23:25.

ORAGE USED: CODE(1) 000317; DATA(0) 000464

    1*          REAL INDEX (30),MU
    2*          DIMENSION E(30),Y(30),W(30),V(30),TITLE(18)
    3*          DATA A1/0.421559/
    4*    C     READ IN EXPERIMENT TITLE
    5*    100   READ (5,5) (TITLE(I),I=1,18)
    6*    5     FORMAT (18A4)
    7*          WRITE (6,6) (TITLE(I),I=1,18)
    8*    6     FORMAT (1H1,10X,18A4/)
    9*    C     READ IN NO. OF DATA POINTS NOB, WEIGHT SWITCH IWGHT (IF 0 ABSOLUTE
   10*    C      ERROR AND W(I) = 1.  IF 1, RELATIVE ERROR AND W(I) = 1/Y(I)**2),
   11*    C     AND REDUCED MASS MU.
   12*    C     NOTE ... IF V IN UNITS OF KM/SEC AND MU IN 1.E-24 GRAM, THEN UNITS
   13*    C     OF ENERGY WILL BE 1.E-14 ERGS. (NOTE THAT CONS HAS UNITS OF ACTION
   14*    C      1.E-27 ERG-SEC)
   15*          READ (5,10) NOB,IWGHT,MU
   16*    10    FORMAT (2I5,F10.4)
   17*          WRITE (6,20) NOB,IWGHT,MU
   18*    20    FORMAT (1HO,10X,'NO. OF OBSERVATIONS ...',I10/11X,'WEIGHTING SWITC
   19*         1H ...',I10/11X,'REDUCED MASS ...',E15.6)
   20*    C     READ IN EXTREMA VELOCITIES AND EXTREMA INDICES
   21*          READ (5,30) (V(I),I=1,NOB)
   22*          READ (5,30) (INDEX(I),I=1,NOB)
   23*    30    FORMAT (8F10.4)
   24*          WRITE (6,40)
   25*    40    FORMAT (////5X,'NO.', 8X,'V',15X,'ETR',11X,'INDEX',11X,'Y'/6X,60(
   26*         1 1H*)/)
   27*          DO 500 I = 1,NOB
   28*          E(I) = 0.5*MU*V(I)*V(I)
   29*          Y(I) = (INDEX(I) - 0.375)*V(I)
   30*          W(I) = 1.
   31*          IF (IWGHT .EQ. 1) W(I) = 1./(Y(I)*Y(I))
   32*    500   WRITE (6,50) I,V(I),E(I),INDEX(I),Y(I)
   33*    50    FORMAT (3X,I5,4E15.6)
   34*          SIG = FIT (E,Y,W,NOB,A1ERM,ERAER,E2R,ERE2R,EPS,EREPS,RM,ERRM)
   35*          WRITE (6,60) SIG
   36*    60    FORMAT (//10X,'RMS RESIDUAL OF FIT ...',E15.5/)
   37*          EPSRM = A1ERM/A1
   38*          EE = ERAER/A1
   39*          WRITE (6,80) A1ERM,ERAER,EPSRM,EE
   40*    80    FORMAT (/10X,'A1*EPS*RM =',E12.5,2X,'(ERROR=',E12.5,' )'/10X,'EPS*
   41*         1RM FROM A1 = 0.421559 ...',E12.5,2X,'(ERROR =',E12.5,' )')
   42*          EPS2 = E2R/EPSRM
   43*          EE = EPS2*SQRT( (ERE2R/E2R)**2 + (EE/EPSRM)**2 )
   44*          WRITE (6,85) E2R,ERE2R,EPS2,EE
   45*    85    FORMAT(/10X,'EPS*EPS*RM =',E12.5,2X,'(ERROR =',E12.5,' )'/10X,'EPS
   46*         1 FROM A1 =0.421559 ...',E12.5,2X,'(LOWER BOUND ON ERROR =',E12.5,
   47*         2 ')' )
   48*          WRITE (6,90) EPS,EREPS
   49*    90    FORMAT (/10X,'EPS =',E12.5,2X,'(ERROR =',E12.5,' )')
   50*          WRITE (6,95) RM,ERRM
   51*    95    FORMAT (/10X,'RM = ',E12.5,2X,'(ERROR =',E12.5,')' )
   52*          GO TO 100
```

```
53*          END

END OF COMPILATION:          NO  DIAGNOSTICS.

        COMPILATION TIME:      0.731 SEC.
        I/O REQUESTS:            51
        I/O WORDS TRANSFERED:  55235
```

A. FIT

This subprogram performs the mechanics of the fit of (V-1), given data E(I), Y(I), W(I), for I = 1, ..., N , where W(I) is the weight of the I-th data point. Initial estimates of c_2 and c_3 (80 and 5 , resp.) are iteratively refined by bilinear regression of the first-order expansion

$$Y_i \simeq c_1 + \left(\frac{\partial Y}{\partial c_2}\right)\Big|_{c_2^\circ, c_3^\circ} + \left(\frac{\partial Y}{\partial c_3}\right)\Big|_{c_2^\circ, c_3^\circ} (c_3 - c_3^\circ) \qquad \text{(V-4)}$$

where

$$\frac{\partial Y}{\partial c_2} = \frac{1}{E}\left(A_1 + \frac{A_2 c_3}{E} + \frac{A_3 c_3^2}{E^2} + \frac{A_4 c_3^3}{E^3}\right)$$

$$\frac{\partial Y}{\partial c_3} = \frac{c_2}{E^2}\left(A_2 + \frac{2A_3 c_3}{E} + \frac{3A_4 c_3^2}{E^2}\right) \qquad \text{(V-5)}$$

The iteration is continued until the rms residual converges to 1%.

Z = FIT(E, Y, W, N, AIERM, ERAEM, E2R, ERE2R, EPS, EREPS, RM, ERRM) returns Z = S the rms residual and

AlERMS = $a_1 \varepsilon r_m$, E2R = $\varepsilon^2 r_m$, EPS = ε , and

ERAEM, ERE2R, ERRM, the resp. 95% confidence interval halfwidths of these quantities.

```
SI FIT
000 COMPILED BY FOR-V S7E6 ON 16 SEP 71 AT 19:23:27.

ORAGE USED: CODE(1) 000304; DATA(0) 000167

    1*          FUNCTION   FIT (E,Y,W,N,AIERM,ERAER,E2R,ERE2R,EPS,EREPS,RM,ERRM)
    2*    C     PERFORMS ITERATIVE FITTING OF EXTREMA VALUES ...
    3*    C     E(I) = TRANSLATIONAL ENERGY OF POINT I
    4*    C     Y(I) =  (N-0.375)*V(I)      --- EXTREMA ORDINATE
    5*          DIMENSION E(1),Y(1),W(1),X(30,2),C(3),ER(4),A(4)
    6*          DATA A/-0.1655,0.1057,-0.0544,0.0139/,CONS/1.65654781/
    7*    C     NOTE - CONS = PI*HCROSS/2.
    8*          EPS = 5.
    9*          E2R = 80.
   10*          SIGP = 0.
   11*          KOUNT = 0
   12*   100    KOUNT = KOUNT + 1
   13*          IF (KOUNT .GT. 10) GO TO 400
   14*          DO 200 I = 1,N
   15*          X(I,1) = (A(1) + EPS *(A(2) + EPS *(A(3) +EPS *A(4)/E(I))/E(I))/
   16*         1 E(I))/E(I)
   17*   200    X(I,2) = E2R *(A(2) +EPS *(2.*A(3) + 3.*EPS *A(4)/E(I))/E(I))/
   18*         1 (E(I)*E(I))
   19*          SIG = BILINR (X,Y,W,N,C,ER,RHO,30)
   20*          EPS = EPS + C(3)
   21*          E2R = C(2)
   22*          DEL = SIG - SIGP
   23*          SIGP = SIG
   24*          IF (ABS(DEL) .GT. 0.01*SIG) GO TO 100
   25*   300    AIERM = CONS*C(1)
   26*          ERAER = CONS*ER(1)
   27*          E2R = CONS*E2R
   28*          ERE2R = CONS*ER(2)
   29*          EREPS = ER(3)
   30*          RM = E2R/(EPS*EPS)
   31*          ERRM = RM*SQRT( 4.*(EREPS/EPS)**2 + (ERE2R/E2R)**2 - 4.*RHO*EREPS
   32*         1*ERE2R/(EPS*E2R) )
   33*          FIT = SIG
   34*          RETURN
   35*   400    WRITE (6,10)
   36*   10     FORMAT (//10X,'*****FAILURE TO CONVERGE IN 10 ITERATIONS'//)
   37*          GO TO 300
   38*          END

END OF COMPILATION:        NO  DIAGNOSTICS.


        COMPILATION TIME:        0.644 SEC.
        I/O REQUESTS:              50
        I/O WORDS TRANSFERED:    54973
```

B. BILINR

Performs a weighted bilinear regression of the model

$$\hat{Y}_i = c_1 + c_2 X_{i1} + c_3 X_{i2}$$

Z = BILINR (X, Y, W, N, C, E, RHO, NDIM)

Input:

X(I,1) - X_{i1}, I-th value of first variable

X(I,2) - X_{i2}, I-th value of second variable

Y(I), W(I) - ordinate and weight of I-th data pt.

N - No. of data

NDIM-row dimension of X

Output:

C(J) - c_j , J-th parameter

E(J) - 95% confidence interval halfwidth of C(J)

RHO - r_{23} correlation coefficient between c_2 and c_3

Z = S rms residual of fit, calculated from

$$S^2 = \frac{1}{N-3} \sum_{i=1}^{N} w_i (Y_i - \hat{Y}_i)^2$$

```
      SIG2 = 0.D0
      SUMY = 0.D0
      DO 200 I = 1,N
      T = Y(I) - C(1) - C(2) *X(I) - C(3)*X(I+NDIM)
      SIG2 = SIG2 + W(I)*T*T
200   SUMY = SUMY + W(I)*T
      C(1) = C(1) + SUMY/SUMW
      IDF = N-3
      SIG2 = SIG2/IDF
      BILINR = DSQRT(SIG2)
      T = T95(IDF)*BILINR
      E(2) = T*DSQRT(A11)
      E(3) = T*DSQRT(A22)
      E(1) = T*DSQRT( (1.D0 + (A11*SUMX1*SUMX1 + 2.D0*A21*SUMX1*SUMX2 +
     1 A22*SUMX2*SUMX2)/SUMW)/SUMW )
      RHO = A21/DSQRT(A11*A22)
      RETURN
      END
```

```
END OF COMPILATION:        NO  DIAGNOSTICS.

        COMPILATION TIME:          1.094 SEC.
        I/O REQUESTS:                 58
        I/O WORDS TRANSFERED:      58883
```

C. EXAMPLES

The four examples are the same as described in IV-B.

1. Note that S = 0.31%
2. $S = 0.12 \times 10^{-5}$ % (true 0%)
3. S = 0.66% (true 0.5%)
4. S = 0.21% (true 0.25%)

DATA OF U. BUCK, K.A. KOHLER, AND H. PAULY , Z. PHYSIK 244 180 (1971)

NO. OF OBSERVATIONS ... 13
WEIGHTING SWITCH ... 1
REDUCED MASS ... .342499+02

NO.	V	ETR	INDEX	Y
1	.234000+01	.937694+02	.500000+01	.108225+02
2	.207500+01	.737336+02	.550000+01	.106344+02
3	.188000+01	.605264+02	.600000+01	.105750+02
4	.170500+01	.497827+02	.650000+01	.104431+02
5	.154500+01	.408777+02	.700000+01	.102356+02
6	.141300+01	.341911+02	.750000+01	.100676+02
7	.130500+01	.291642+02	.800000+01	.995062+01
8	.120700+01	.249485+02	.850000+01	.980681+01
9	.112500+01	.216738+02	.900000+01	.970312+01
10	.104500+01	.187009+02	.950000+01	.953562+01
11	.968000+00	.160465+02	.100000+02	.931700+01
12	.910000+00	.141812+02	.105000+02	.921375+01
13	.855000+00	.125188+02	.110000+02	.908437+01

RMS RESIDUAL OF FIT 30755-02

A1*EPS*RM = .18748+02 (ERROR= .13757+00)
EPS*RM FROM A1 = 0.421559 ... .44474+02 (ERROR = .32633+00)

EPS*EPS*RM = .56106+03 (ERROR = .57075+02)
EPS FROM A1 =0.421559 ... .12615+02 (LOWER BOUND ON ERROR = .12867+01)

EPS = .18831+02 (ERROR = .24142+01)

RM = .15822+01 (ERROR = .26027+00)

TEST DATA --- NEW TERMS -- NO ERROR -- EPS = 8. -- RM = 4.

NO. OF OBSERVATIONS ... 13
WEIGHTING SWITCH ... 1
REDUCED MASS ... .300000+02

NO.	V	ETR	INDEX	Y
1	.800000+00	.960000+01	.839812+01	.641849+01
2	.900000+00	.121500+02	.779061+01	.667406+01
3	.100000+01	.150000+02	.725640+01	.688140+01
4	.110000+01	.181500+02	.678505+01	.705106+01
5	.120000+01	.216000+02	.636754+01	.719105+01
6	.130000+01	.253500+02	.599611+01	.730744+01
7	.140000+01	.294000+02	.566423+01	.740492+01
8	.150000+01	.337500+02	.536643+01	.748714+01
9	.160000+01	.384000+02	.509810+01	.755695+01
10	.170000+01	.433500+02	.485537+01	.761663+01
11	.180000+01	.486000+02	.463497+01	.766795+01
12	.190000+01	.541500+02	.443414+01	.771236+01
13	.200000+01	.600000+02	.425050+01	.775100+01

RMS RESIDUAL OF FIT 12267-07

AI*EPS*RM = .13490+02 (ERROR= .59740-06)
EPS*RM FROM AI = 0.421559 ... .32000+02 (ERROR = .14171-05)

EPS*EPS*RM = .25600+03 (ERROR = .21102-03)
EPS FROM AI =0.421559 ... .80000+01 (LOWER BOUND ON ERROR = .66039-05

EPS = .80000+01 (ERROR = .15783-04)

RM = .40000+01 (ERROR = .12538-04)

TEST DATA --- EPS = 8. --- RM = 4. --- SIGMA = 0.005

NO. OF OBSERVATIONS ... 13
WEIGHTING SWITCH ... 1
REDUCED MASS ... .300000+02

NO.	V	ETR	INDEX	Y
1	.800000+00	.960000+01	.826601+01	.631281+01
2	.900000+00	.121500+02	.780906+01	.669065+01
3	.100000+01	.150000+02	.721831+01	.684331+01
4	.110000+01	.181500+02	.674615+01	.700827+01
5	.120000+01	.216000+02	.628429+01	.709115+01
6	.130000+01	.253500+02	.598548+01	.729362+01
7	.140000+01	.294000+02	.568245+01	.743044+01
8	.150000+01	.337500+02	.531429+01	.740894+01
9	.160000+01	.384000+02	.511698+01	.758717+01
10	.170000+01	.433500+02	.487927+01	.765726+01
11	.180000+01	.486000+02	.464704+01	.768967+01
12	.190000+01	.541500+02	.446258+01	.776641+01
13	.200000+01	.600000+02	.426992+01	.778986+01

RMS RESIDUAL OF FIT 66103-02

A1*EPS*RM = .13653+02 (ERROR= .32379+00)
EPS*RM FROM A1 = 0.421559 ... .32386+02 (ERROR = .76808+00)

EPS*EPS*RM = .30311+03 (ERROR = .12063+03)
EPS FROM A1 =0.421559 ... .93591+01 (LOWER BOUND ON ERROR = .37312+01)

EPS = .98608+01 (ERROR = .79641+01)

RM = .31173+01 (ERROR = .38182+01)

TEST DATA --- EPS = 8. --- RM = 4. --- SIGMA = 0.0025

NO. OF OBSERVATIONS ... 13
WEIGHTING SWITCH ... 1
REDUCED MASS ... .300000+02

NO.	V	ETR	INDEX	Y
1	.800000+00	.960000+01	.838536+01	.640828+01
2	.900000+00	.121500+02	.780310+01	.668529+01
3	.100000+01	.150000+02	.727100+01	.689600+01
4	.110000+01	.181500+02	.680110+01	.706871+01
5	.120000+01	.216000+02	.636197+01	.718436+01
6	.130000+01	.253500+02	.598382+01	.729147+01
7	.140000+01	.294000+02	.566941+01	.741217+01
8	.150000+01	.337500+02	.535622+01	.747183+01
9	.160000+01	.384000+02	.511618+01	.758589+01
10	.170000+01	.433500+02	.485833+01	.762166+01
11	.180000+01	.486000+02	.464257+01	.768163+01
12	.190000+01	.541500+02	.443228+01	.770883+01
13	.200000+01	.600000+02	.425656+01	.776312+01

RMS RESIDUAL OF FIT 20544-02

A1*EPS*RM = .13495+02 (ERROR= .98363-01)
EPS*RM FROM A1 = 0.421559 ... .32013+02 (ERROR = .23333+00)

EPS*EPS*RM = .25410+03 (ERROR = .34287+02)
EPS FROM A1 =0.421559 ... .79372+01 (LOWER BOUND ON ERROR = .10726+01)

EPS = .77712+01 (ERROR = .25466+01)

RM = .42075+01 (ERROR = .21990+01)

VI. QFIT

This main program directs a weighted nonlinear regression of the model

$$\hat{Q}(v) = \gamma_1 v^{-2/5} + \gamma_2 v^{1/2}\left[1 + \frac{H_1 \gamma_5}{E}\right] \sin\phi \qquad \text{(VI-1)}$$

where

$$\phi = -\frac{3\pi}{4} + \frac{\gamma_3}{v} + \frac{\gamma_4}{vE}\left(A_1 + \frac{A_2\gamma_5}{E} + \frac{A_3\gamma_5^2}{E^2} + \frac{A_4\gamma_5^3}{E^3}\right) \qquad \text{(VI-2)}$$

and after regression the ansatz $\gamma_5 = \epsilon$ and (I-8) is made, leading to

$$\begin{aligned} C_6 &= \hbar\,(\gamma_1/8.083)^{5/2} \\ r_m &= \hbar\gamma_4/4\gamma_5^2 \\ a_1 &= \gamma_3\gamma_5/\gamma_4 \\ g_0 &= \gamma_5^{7/2}\gamma_2/\hbar^2(\pi\gamma_4/2)^{3/2} \\ \epsilon &= \gamma_5 \end{aligned} \qquad \text{(VI-3)}$$

The regression is automatically repeated with initial values for $\gamma_1 - \gamma_5$ being the values obtained from the first regression, and a new adjustable parameter γ_6 being added to the model:

$$\hat{Q}(v) = \gamma_1 v^{-2/5} + \gamma_2 v^{1/2}\left[1 + \frac{H_1\gamma_5}{E}\right]\sin\phi + \gamma_6 \qquad \text{(VI-4)}$$

After regression the ansatz

$$\gamma_6 = \frac{2\pi}{3} \frac{C_8}{C_6} \qquad \text{(VI-5)}$$

is made, leading to

$$C_8 = \frac{3\hbar}{2\pi} \gamma_6 \left(\frac{\gamma_1}{8.083}\right)^{5/2} \qquad \text{(VI-6)}$$

The required input is of the following form:

CARD 1: Title information (18A4)

CARD 2: N, MU, EPS, RM, A1, GO, C6, IWGHT (I5,5F10.4,E15.6,I5)

CARD 3 (or more): V(I), I = 1, ..., N (8F10.4)

CARD 4 (or more): Q(I), I = 1, ..., N (8F10.4)

where

N - Number of velocities at which cross sections are given

MU - μ , reduced mass of the system

EPS, RM, A1, GO, C6 - initial estimates for ε , r_m, a_1 , g_o , C_6

IWGHT - if 0 data are assumed to have constant absolute error; if 1, constant relative error

V(I) - the I-th velocity observation

Q(I) - the I-th cross section observation.

If the scale of the Q's is not known, C_6, C_8 and g_o will be found finally to contain the same factor, but no other parameters will be affected.

Output of the program includes the values of $\gamma_1, \ldots, \gamma_6$ with error intervals (denoted parameters 1-6), and the correlation matrix of the errors of $\gamma_1, \ldots, \gamma_6$.

```
I GFIT
000 COMPILED BY FOR-V S7E6 ON 14 SEP 71 AT 09:22:29.

RAGE USED: CODE(1) 000272; DATA(0) 007340; BLANK COMMON(2) 000620

    1*    C     A CONSISTENT SET OF UNITS FOR THIS PROGRAM IS ...
    2*    C     1.E-24 GM FOR MASS, 1.E-14 ERGS FOR ENERGY, ANGSTROMS FOR LENGTH
    3*    C     AND KM/SEC FOR VELOCITY.
    4*          REAL MU
    5*          DOUBLE PRECISION R(36),X,PAR,ER,F(200),F2(200),BUP(6),BLOW(6)
    6*          DIMENSION X(1200),Q(200),W(200),PAR(6),ER(6)
    7*          COMMON V(200),E(200)
    8*          DATA BLOW/6*0.00/,BUP/1.D8,1.D2,5.D1,2.D4,1.D4,1.D8/
    9*          EXTERNAL XSECT,DXSECT
   10*    100   READ (5,10) (Q(I),I=1,18)
   11*    10    FORMAT (18A4)
   12*          WRITE (6,20) (Q(I),I=1,18)
   13*    20    FORMAT (1H1,10X,18A4)
   14*          READ (5,30) N,MU,EPS,RM,A1,GO,C6,IWGHT
   15*    C     IF IWGHT = 0, ASSUMED CONSTANT ERROR. IF IWGHT= 1, ASSUMED
   16*    C      CONSTANT RELATIVE ERROR.
   17*    C     FOR A LENNARD-JONES 12-6 POTENTIAL A1 = 0.421559, GO = 0.188299
   18*    C     CRUDE INITIAL ESTIMATES FOR EPS AND RM MIGHT BE 5. AND 4., RESP.
   19*    30    FORMAT (I5,5F10.4,E15.6,I5)
   20*          WRITE (6,35) N,MU,EPS,RM,A1,GO,C6,IWGHT
   21*    35    FORMAT (1HO,10X,'NO OF DATA POINTS',I8/11X,'REDUCED MASS',E15.6/
   22*         1 11X,'INITIAL EPS',E15.6/11X,'INITIAL RM',E15.6/11X,'INITIAL A1',
   23*         2 E15.6/11X,'INITIAL GO',E15.6/11X,'INITIAL C6',E15.6/11X,'WEIGHTIN
   24*         3G SWITCH',I8/)
   25*          CALL TRIAL (PAR,C6,A1,GO,EPS,RM)
   26*          READ (5,40) (V(I),I=1,N)
   27*          READ (5,40) (Q(I),I=1,N)
   28*    40    FORMAT (8F10.4)
   29*          DO 200 I = 1,N
   30*          W(I) = 1.
   31*          IF (IWGHT .EQ. 1) W(I) = 1./(Q(I)*Q(I))
   32*    200   E(I) = 0.5*MU*V(I)*V(I)
   33*          WRITE (6,60)
   34*    60     FORMAT (//'0    FIT WITH 5 PARAMETERS (NO C8)'//)
   35*          CALL GASS59 (1,XSECT,DXSECT,N,W,Q,5,PAR,BUP,BLOW,1.D-3,1.D-3,1.D-6
   36*         1, 20,R,ER,F,F2,X)
   37*          CALL FINAL (5,PAR,ER,R)
   38*          WRITE (6,70)
   39*    70    FORMAT (//'0    FIT WITH 6 PARAMETERS (WITH C8)'//)
   40*          CALL GASS59 (1,XSECT,DXSECT,N,W,Q,6,PAR,BUP,BLOW,1.D-3,1.D-3,1.D-6
   41*         1, 10,R,ER,F,F2,X)
   42*          CALL FINAL (6,PAR,ER ,R)
   43*          GO TO 100
   44*          END

END OF COMPILATION:        NO  DIAGNOSTICS.


        COMPILATION TIME:          0.664 SEC.
        I/O REQUESTS:                51
        I/O WORDS TRANSFERED:      55129
```

A. TRIAL

This subprogram converts initial estimates for ε , r_m , a_1 , g_o and C_6 into initial estimates for γ_1 , ... , γ_6 (denoted by PAR(1) ... PAR(6).)

CALL TRIAL (PAR, C6, A1, GO, EPS, RM)

Input:

C6 - long-range C_6

A1 - a_1 (I-5c)

GO - g_o (I-5a)

EPS - ε

RM - r_m

Output:

PAR(1) = γ_1 (I-8)

PAR(2) = γ_2 (I-8)

PAR(6) = γ_6 (VI-8)

```
I TRIAL
000 COMPILED BY FOR-V S7E6 ON 14 SEP 71 AT 09:22:46.

RAGE USED: CODE(1) 000102; DATA(0) 000035

    1*          SUBROUTINE TRIAL (PAR,C6,A1,GO  ,EPS,RM)
    2*          DOUBLE PRECISION PAR
    3*          DIMENSION PAR(1)
    4*          DATA HCROSS/1.0545919/,PI/3.14159265/
    5*          PAR(1) = 8.083*(C6/HCROSS)**0.4
    6*          PAR(2) = (2.*PI*RM)*GO  *SQRT(2.*PI*RM*HCROSS/EPS)
    7*          PAR(3) = EPS
    8*          PAR(4) = 4.*RM*EPS/HCROSS
    9*          PAR(5) = EPS*PAR(4)
   10*          PAR(4) = A1*PAR(4)
   11*          PAR(6) = 0.
   12*          RETURN
   13*          END

END OF COMPILATION:          NO  DIAGNOSTICS.

        COMPILATION TIME:          0.395 SEC.
        I/O REQUESTS:                 49
        I/O WORDS TRANSFERED:      52841
```

B. FINAL

This subprogram takes the final values of γ_1 , ... , γ_6 and their errors, and outputs values and errors for ε , a_1 , r_m , C_6 , g_o , $a_1 \varepsilon r_m$, $\varepsilon^2 r_m$, β , and C_8 .

CALL FINAL (NP, PAR, ERR, R)

where:

NP - no. of parameters of fit (5 or 6)

PAR(I) - γ_i $1 \leq I \leq NP$

ERR(I) - 95% confidence interval halfwidth for PAR(I)

R(I,J) - correlation coefficient r_{ij} between PAR(I) and PAR(J), stored as an NPxNP matrix.

```
I FINAL
000 COMPILED BY FOR-V S7E6 ON 14 SEP 71 AT 09:22:49.

RAGE USED: CODE(1) 000540; DATA(0) 000177

    1*          SUBROUTINE FINAL (NP,PAR,ERR,R)
    2*          DOUBLE PRECISION PAR,ERR,R(1)
    3*          DIMENSION PAR(1),ERR(1)
    4*          DATA HCROSS/1.0545919/,PI/3.14159265/
    5*          NN = NP + 1
    6*          EPS = PAR(3)
    7*          EREPS = ERR(3)
    8*          A1 = PAR(4)*EPS/PAR(5)
    9*          ERA1 = A1*SQRT((ERR(4)/PAR(4))**2 + (ERR(3)/PAR(3))**2 + (ERR(5)/
   10*         1 PAR(5))**2 + 2.*R(3+3*NN)*ERR(3)*ERR(4)/(PAR(3)*PAR(4)) -2.*
   11*         2 R(3+4*NN)*ERR(3)*ERR(5)/(PAR(3)*PAR(5)) -2.*R(4+4*NN)*ERR(4)*
   12*         3 ERR(5)/(PAR(4)*PAR(5)) )
   13*          RM = 0.25*PAR(5)*HCROSS/(PAR(3)*PAR(3))
   14*          ERRM = RM*SQRT( 4.*(ERR(3)/PAR(3))**2 + (ERR(5)/PAR(5))**2 -4.*
   15*         1 R(3+4*NN)*ERR(3)*ERR(5)/(PAR(3)*PAR(5)) )
   16*          C6 = HCROSS*(PAR(1)/8.083)**2.5
   17*          ERC6 = 2.5*C6*ERR(1)/PAR(1)
   18*          GO = 8.*PAR(2)*PAR(3)**3.5/(HCROSS*HCROSS*(2.*PI*PAR(5))**1.5)
   19*          ERGO = GO*SQRT((ERR(2)/PAR(2))**2+12.25*(ERR(3)/PAR(3))**2 +2.25*
   20*         1 (ERR(5)/PAR(5))**2  + 7.*R(2+2*NN)*ERR(2)*ERR(3)/(PAR(2)*PAR(3))
   21*         2 -3.*R(2+4*NN)*ERR(2)*ERR(5)/(PAR(2)*PAR(5))-10.5*R(3+4*NN)*ERR(3)
   22*         3 *ERR(5)/(PAR(3)*PAR(5)) )
   23*          WRITE (6,10)
   24*   10     FORMAT (///10X,'TERM',8X,'VALUE',8X,'95 PERCENT HALFWIDTH'/5X,
   25*         1 50(1H*)/)
   26*          WRITE (6,20) C6,ERC6,GO,ERGO,EPS,EREPS,A1,ERA1,RM,ERRM
   27*   20     FORMAT (10X,' C6 ',2E15.6/10X,' GO ',2E15.6/10X,' EPS',2E15.6/10X,
   28*         1 ' A1 ',2E15.6/10X,' RM ',2E15.6)
   29*          A1ERM = 0.25*HCROSS *PAR(4)
   30*          ERAER = 0.25*HCROSS*ERR(4)
   31*          WRITE (6,30) A1ERM,ERAER
   32*   30     FORMAT (/8X,'A1*EPS*RM',E12.6,E15.6)
   33*          E2R = 0.25*HCROSS*PAR(5)
   34*          ERE2R = 0.25*HCROSS*ERR(5)
   35*          WRITE (6,35) E2R,ERE2R
   36*   35     FORMAT (/8X,'RM*EPS**2',E12.6,E15.6)
   37*          IF (NP .LT. 6) RETURN
   38*          BETA = 1.5*PAR(6)/PI
   39*          ERBETA = 1.5*ERR(6)/PI
   40*          WRITE (6,40) BETA,ERBETA
   41*   40     FORMAT (/10X,'BETA',2E15.6)
   42*          C8 = C6*BETA
   43*          ERC8 = C8*SQRT((ERC6/C6)**2 + (ERBETA/BETA)**2 +2.*R(1+5*NN)*ERC6
   44*         1 *ERBETA/(C6*BETA))
   45*          WRITE (6,50) C8,ERC8
   46*   50     FORMAT (/10X,' C8 ',2E15.6)
   47*          RETURN
   48*          END

END OF COMPILATION:        NO  DIAGNOSTICS.
```

C. XSECT/DXSECT

Subprogram which calculates $Q(v)$ given $\gamma_1, \ldots, \gamma_6$ by means of (I-6) (XSECT), or $\partial Q(v)/\partial\gamma_i$, $1 \leq i \leq 6$ (DXSECT).

1. CALL XSECT (PAR, F, NOB, NP)

   Input:

   PAR(I) - γ_i $\quad 1 \leq I \leq NP$

   NOB - No. of velocities v

   NP - No. of parameters γ_i (= 5 or 6)

   Output:

   F(I) - $Q(v)$ for I-th velocity

2. CALL DXSECT (PAR, F, NOB, NP)

   Input:

   PAR(I), NOB, NP - as above

   Output:

   F(I,J) - (NOB X NP matrix) $\partial Q(V)/\partial\gamma_j$ for I-th velocity

```
I XSECT
000 COMPILED BY FOR-V S7E6 ON 14 SEP 71 AT 09:22:56.

RAGE USED: CODE(1) 000420; DATA(0) 000110; BLANK COMMON(2) 000620

    1*          SUBROUTINE XSECT (PAR,F,NOB,NP)
    2*    C     SUBROUTINE WHICH CALCULATES CROSS SECTIONS GIVEN VALUES
    3*    C     OF THE FOLLOWING PARAMETERS ...
    4*    C     PAR(1) = 8.083*(C6/HCROSS)**0.4
    5*    C     PAR(2) = 2.*PI*RM*GO*SQRT(2.*PI*RM*HCROSS/EPS)
    6*    C     PAR(3) = EPS
    7*    C     PAR(4) = 4.*A1*EPS*RM/HCROSS
    8*    C     PAR(5) = 4.*EPS*EPS*RM/HCROSS
    9*    C     PAR(6) = 2.*PI*C8/(C6*3.)
   10*          DOUBLE PRECISION PAR,F,A,H1,PI75,PHI,Z
   11*          DIMENSION PAR(1),F(1),A(4)
   12*          COMMON V(200),E(200)
   13*          DATA A/-0.1655D0,0.1057D0,-0.5440-1,0.1390-1/
   14*          DATA H1/3.2670D0/,PI75/2.3561944900/
   15*          DO 100 I = 1,NOB
   16*          Z = PAR(3)/E(I)
   17*          PHI = -PI75 + (PAR(4) + PAR(5)*(A(1) +Z*(A(2) +Z*(A(3) +Z*A(4))))/
   18*         1 E(I))/V(I)
   19*   100    F(I) = PAR(1)/V(I)**0.4 + PAR(2)*DSQRT(V(I))*(1.DD +H1*Z)*DSIN(PHI
   20*         1) + PAR(6)
   21*          RETURN
   22*          ENTRY DXSECT (PAR,F,NOB,NP)
   23*    C     ENTRY POINT WHICH RETURNS MATRIX OF DERIVATIVES OF MODEL WITH
   24*    C     RESPECT TO PARAMETERS
   25*          DO 500 I = 1,NOB
   26*          F(I) = 1.DD/V(I)**0.4
   27*          Z = PAR(3)/E(I)
   28*          SUM = (A(1) + Z*(A(2) + Z*(A(3) + Z*A(4))))/E(I)
   29*          PHI = -PI75 + (PAR(4) + PAR(5)*SUM)/V(I)
   30*          SPHI = DSIN(PHI)
   31*          CPHI = DCOS(PHI)
   32*          ROOTV = SQRT(V(I))
   33*          TERM = 1.DD + H1*Z
   34*          PROD = ROOTV*TERM
   35*          F(I+NOB) = PROD*SPHI
   36*          F(I+2*NOB) = PAR(2)*ROOTV*(TERM*CPHI*PAR(5)*(A(2) + Z*(2.*A(3) +
   37*         1 3.*Z*A(4)))/(E(I)*V(I)) + H1*SPHI)/E(I)
   38*          PROD = PROD*PAR(2)*CPHI
   39*          F(I+3*NOB) = PROD/V(I)
   40*          IF (NP .EQ. 6) F(I+5*NOB) = 1.DD
   41*   500    F(I+4*NOB) = PROD*SUM/V(I)
   42*          RETURN
   43*          END

END OF COMPILATION:        NO  DIAGNOSTICS.

        COMPILATION TIME:          0.978 SEC.
        I/O REQUESTS:                59
        I/O WORDS TRANSFERED:     57954
```

D. GASS59/MAXLIK

This subprogram performs general nonlinear regression of NOB observations Y(I), with weights W(I), to the model

$$\hat{Y} = \hat{Y}(\theta_1, \cdots, \theta_n)$$

This program is a reworked version of a program of the same name of the Univ. Wis. Computing Center.

1. CALL GASS59(IO, YMU, DYMU, NOB, W, Y, NP, TH, BUP, BLOW, EPS1, EPS2, EPS3, MAXIT, D, E, F, R, DELZ)

INPUT: IO - I/O parameter. If 0, no printing is done. If 1, only last iteration is printed. If 2, all printing is performed.

CALL YMU (TH, F, NOB, NP) - stores $\hat{Y}(I)$ in F(I) $1 \leq I \leq NOB$

CALL DYMU (TH, DELZ, NOB, NP) - Stores $\partial\hat{Y}(I)/\partial\theta_j$ in DELZ(I,J) $1 \leq I \leq NOB$, $1 \leq J \leq NP$.

NOB - No. of observations

Y(I) - I-th observed value

NP - n, No. of parameters $\theta_1, \ldots \theta_n$

TH(J) - J-th parameter value θ_j . Initially should be set to an estimate of θ_j .

BUP(J), BLOW(J) - Bounds set upon the allowable values of TH(J): BLOW(J) $\leq$ TH(J) $\leq$ BUP(J).

EPS1 - Stopping criterion for relative change in the sum of squared residuals (SSR) per iteration.

EPS2 - Stopping criterion for relative change in each parameter per iteration.

EPS3 - Stopping criterion for relative change in SSR from initial SSR.

MAXIT - Maximum allowed number of iterations (e.g. 20)

F,R - work vectors of size NOB

DELZ - work space of size NOB X NP

OUTPUT: TH(I) - determined value of θ_i

D(I,J) - (NP X NP matrix) determined value of correlation between θ_i and θ_j

E(I) - determined 95% confidence interval halfwidth for θ_i

2. CALL MAXLIK (IO, YMU, DYMU, NOB, W, Y, NP, TH, D, E, F, R, DELZ)

The iterative fitting of the θ vector is done between entry points GASS59 and MAXLIK; from entry point MAXLIK onwards, the statistics of the fit are determined. A call to MAXLIK calculates the statistics for the given θ vector, but does not determine the θ vector by means of a fit.

```
I GASS59
000 COMPILED BY FOR-V S7E6 ON 14 SEP 71 AT 09:23:02.

RAGE USED: CODE(1) 003321; DATA(0) 002626

    1*          SUBROUTINE GASS59 (IO,YMU,DYMU,NOB,W,Y,NP,TH,BUP,BLOW,EPS1,EPS2,
    2*         1 EPS3,MAXIT,D,E,F,R,DELZ)
    3*          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
    4*          REAL Y,W
    5*          REAL STUDNT
    6*          DIMENSION Y(1),W(1),TH(1),BUP(1),BLOW(1),D(1),E(1),Q(20),A(400),
    7*         1 P(20),PHI(20),TB(20),WTS(20),F(1),R(1),DELZ(1)
    8*          DATA FLAM/1.D-2/,FNU/1.D1/
    9*          NPSQ = NP*NP
   10*          GA = FLAM
   11*          IF (IO .EQ. 0) GO TO 50
   12*          WRITE (6,1) NOB,NP
   13*    1     FORMAT ('1 ENTRY TO GASS59'//10X,'FIT TO',I8,3X,'OBSERVATIONS'/10X
   14*         1,'THERE ARE',I8,3X,'PARAMETERS IN THE MODEL'/)
   15*          WRITE (6,2)
   16*    2     FORMAT (//5X,'INITIAL PARAMETER VALUES'//15X,'LOWER BOUND',9X,'PAR
   17*         1AMETER VALUE',5X,'UPPER BOUND'/5X,70(1H-)//)
   18*          WRITE (6,2001) (I,BLOW(I),TH(I),BUP(I),I=1,NP)
   19*    50    ITER = 1
   20*          SSQ = 0.D0
   21*          CALL YMU(TH,F,NOB,NP)
   22*          DO 100 I = 1,NOB
   23*          R(I) = Y(I) - F(I)
   24*    100   SSQ = SSQ + W(I)*R(I)*R(I)
   25*          KOUNT = 1
   26*          SSQINT = SSQ
   27*          SSQB4 = SSQ
   28*          IF (IO .NE. 0) WRITE (6,3) SSQ
   29*    3     FORMAT (//5X,'INITIAL SUM OF SQUARES',D20.10)
   30*          IF (SSQ .LT. 1.D-50) RETURN
   31*    150   CONTINUE
   32*          IF (IO .GE. 2) WRITE (6,4) ITER,KOUNT
   33*    4     FORMAT (///5X,'AT ITERATION',I5,10X,'CUMULATIVE NO. FUNCTION CALLS
   34*         1 MADE',I10)
   35*          GA = GA*DSQRT(SSQ/SSQB4)/FNU
   36*          SSQB4 = SSQ
   37*          INTCNT = 0
   38*          CALL DYMU(TH,DELZ,NOB,NP)
   39*          NI = - NOB
   40*          DO 650 I = 1,NP
   41*          NI = NI + NOB
   42*          II = (I-1)*NP
   43*          IJ = I - NP
   44*          NJ = - NOB
   45*          DO 550 J = 1,I
   46*          NJ = NJ + NOB
   47*          IJ = IJ + NP
   48*          JI = II + J
   49*          SUMD = 0.D0
   50*          DO 500 K = 1,NOB
   51*    500   SUMD = SUMD + DELZ(NI+K)*DELZ(NJ+K)*W(K)
   52*          D(JI) = SUMD
```

```
      550   D(IJ) = SUMD
            Q(I) = 0.DO
            DO 600 K = 1,NOB
      600   Q(I) = Q(I) + W(K)*DELZ(NI+K)*R(K)
            II = II + I
            IF (D(II) .LT. 1.D-50) D(II) = 1.D-50
      650   E(I) = DSQRT(D(II))
      700   NI = - NP
            DO 710 I = 1,NP
            NI = NI + NP
            IJ = I - NP
            WTS(I) = 0.DO
            DO 710 J = 1,I
            IJ = IJ + NP
            JI = NI + J
            TEMP = D(IJ)
            A(IJ) = TEMP
      710   A(JI) = TEMP
            CALL SYMQR (A,NP,NP,P,DELZ)
            NI = -NP
            DO 730 I = 1,NP
            NI = NI + NP
            DO 730 J = 1,NP
            JI = NI + J
      730   WTS(J) = WTS(J) + DABS(DELZ(JI))*P(I)
            MASTER = 1
      760   DPROD = 1.DO
            NI = - NP
            DO 770 I = 1,NP
            NI = NI + NP
            IF (WTS(I) .GT. WTS(MASTER)) MASTER = I
            DPROD = DPROD*D(NI+I)**(1./NP)
            IJ = I - NP
            DO 770 J = 1,I
            IJ = IJ + NP
            JI = NI + J
            A(IJ) = D(IJ)/(E(I)*E(J))
      770   A(JI) = A(IJ)
            IF (DPROD .LT. 1.D-50) DPROD = 1.D-50
            WTSMST = WTS(MASTER)
            DO 780 I = 1,NP
            II = (I-1)*NP + I
            P(I) = Q(I)/E(I)
            PHI(I) = P(I)
            IF (DABS(WTS(I)) .LT. 1.D-50) WTS(I) = 1.D-50
            WTS(I) = WTSMST/WTS(I)
      780   A(II) = A(II) + GA*WTS(I)*DPROD/D(II)
            CALL MATIN (A,NP,P,1,DET,PIVRAT)
            IF (PIVRAT .LT. 1.D-16) GO TO 990
            STEP = 1.DO
            SUM1 = 0.DO
            SUM2 = 0.DO
            SUM3 = 0.DO
            DO 790 I = 1,NP
            SUM1 = P(I)*PHI(I) + SUM1
            SUM2 = SUM2 + P(I)*P(I)
            SUM3 = SUM3 + PHI(I)*PHI(I)
```

```
110*    790   PHI(I) = P(I)
111*          TEMP = SUM1/DSQRT(SUM2*SUM3)
112*          IF (DABS(TEMP) .GT. 1.DO) TEMP = 1.DO
113*          TEMP = 57.29578DO*DACOS(TEMP)
114*          IF (IO .LT. 2) GO TO 800
115*          WRITE (6,6) DET,PIVRAT,TEMP
116*    6     FORMAT (/9X,'DETERMINANT',D20.10,5X,'RATIO OF PIVOTS',D20.10/
117*         19X,'ANGLE IN SCALED COORDINATES',F10.5)
118*    800   DO 810 I = 1,NP
119*    810   P(I) = PHI(I)*STEP/E(I)
120*          ISW = 0
121*          DO 820 I = 1,NP
122*          TB(I) = TH(I) + P(I)
123*    820   IF (TB(I) .GT. BUP(I) .OR. TB(I) .LT. BLOW(I)) ISW = 1
124*          IF (IO .LT. 2) GO TO 830
125*          WRITE (6,7)
126*    7     FORMAT (/9X,'TEST POINT PARAMETER VALUES')
127*          WRITE (6,2011) (TB(I),I=1,NP)
128*    830   IF (ISW .EQ. 1) GO TO 860
129*          SUMB = 0.DO
130*          CALL YMU(TB,F,NOB,NP)
131*          KOUNT = KOUNT + 1
132*          DO 840 I = 1,NOB
133*          R(I) = Y(I) - F(I)
134*    840   SUMB = SUMB + W(I)*R(I)*R(I)
135*          IF (IO .GT. 1) WRITE (6,8) SUMB
136*    8     FORMAT (/9X,'TEST POINT SUM OF SQUARES',D20.10)
137*    850   IF (SUMB .LE. (1.DO + EPS1)*SSQ) GO TO 890
138*    860   IF (TEMP .GT. 3.D1 .AND. GA .GT. 1.D-50) GO TO 880
139*    870   STEP = 0.5DO*STEP
140*          INTCNT = INTCNT + 1
141*          IF (INTCNT .GE. 36) GO TO 980
142*          GO TO 800
143*    880   GA = GA*FNU
144*          INTCNT = INTCNT + 1
145*          IF (INTCNT .GE. 36) GO TO 980
146*          GO TO 760
147*    890   DO 900 I = 1,NP
148*    900   TH(I) = TB(I)
149*          SSG = SUMB
150*          IF (IO .GE. 2) WRITE (6,10) GA
151*    10    FORMAT (/9X,'AT THE END OF THE REGRESSION, GAMMA IS',D20.10)
152*          IF (EPS2 .LE. 0.DO) GO TO 920
153*          DO 910 I = 1,NP
154*          IF (DABS(P(I)) .GT. DABS(TH(I)*EPS2)) GO TO 920
155*    910   CONTINUE
156*          WRITE (6,11) EPS2
157*    11    FORMAT (/'0ITERATION STOPS ... RELATIVE CHANGE IN EACH PARAMETER I
158*         1S LESS THAN',D20.10)
159*          GO TO 999
160*    920   IF (DABS(SUMB -SSQB4) .GT. EPS1*SSQB4) GO TO 930
161*          WRITE (6,12) EPS1
162*    12    FORMAT (/' ITERATION STOPS ... RELATIVE CHANGE IN SUM OF SQUARES I
163*         1S LESS THAN',D20.10)
164*          GO TO 999
X *DIAGNOSTIC* THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
165*    930   IF (STEP .NE. 1.DO) GO TO 970
```

```
      DO 960 IDUB = 1,5
      DO 935 I = 1,NP
      TB(I) = TH(I) + P(I)
      IF (TB(I) .GT. BUP(I) .OR. TB(I) .LT. BLOW(I)) GO TO 970
  935 CONTINUE
      SUMB = 0.DO
      CALL YMU(TB,DELZ,NOB,NP)
      KOUNT = KOUNT + 1
      DO 940 I = 1,NOB
  940 SUMB = SUMB + W(I)*(Y(I) - DELZ(I))**2
      IF ((1.DO + EPS1)*SUMB .GE. SSQ) GO TO 970
      DO 945 I = 1,NP
  945 TH(I) = TB(I)
      DO 950 I = 1,NOB
      R(I) = Y(I) - DELZ(I)
  950 F(I) = DELZ(I)
      IF (IO .LT. 2) GO TO 960
      WRITE (6,7)
      WRITE (6,2011) (TB(I) ,I=1,NP )
      WRITE (6,8) SUMB
  960 SSQ = SUMB
  970 CONTINUE
      ITER = ITER + 1
      IF (SSQ .LE. EPS3*SSQINT) GO TO 991
      IF (ITER - MAXIT) 150,150,992
  980 WRITE (6,13)
   13 FORMAT (///'0****SUM OF SQUARES CANNOT BE REDUCED TO SUM OF SQUARE
     1S AT LAST ITERATION-TERMINATION OCCURS'/)
      GO TO 999
  990 WRITE(6,14) DET,PIVRAT
   14 FORMAT(/'ORATIO OF PIVOTS IS TOO SMALL'/' DETERMINANT ',5X,D20.10,
     15X,'RATIO OF PIVOTS',D20.10)
      GO TO 999
  991 WRITE (6,15) EPS3
   15 FORMAT (/'0ITERATION STOPS ... CHANGE IN SUM OF SQUARES (RELATIVE
     1TO INITIAL VALUE) IS LESS THAN',D20.10)
      GO TO 999
  992 WRITE (6,16) MAXIT
   16 FORMAT (/' *****ITERATION TERMINATES-- NO. OF ITERATIONS EXCEEDS',
     1I10)
 2001 FORMAT (I9,3D20.10)
 2011 FORMAT (5X,6D20.10)
 2012 FORMAT(/I10,6D20.10/(10X,6D20.10))
      GO TO 999
      ENTRY MAXLIK (IO,YMU,DYMU,NOB,W,Y,NP,TH,D,E,F,R,DELZ)
      NPSQ = NP*NP
  999 SUMW = 0.DO
      CALL YMU(TH,F,NOB,NP)
      SSQ = 0.DO
      IF (IO .GT. 1) WRITE (6,1011)
      DO 1000 I = 1,NOB
      SUMW = SUMW + W(I)
      R(I) = Y(I) - F(I)
      IF (IO .LE. 1) GO TO 1000
      TEMP = DSQRT(W(I))*R(I)
      WRITE (6,1019) I,Y(I),F(I),R(I),TEMP
 1000 SSQ = SSQ + W(I)*R(I)*R(I)
```

```
 1011  FORMAT (1H1,5X,'NO.',8X,'OBSERVED',7X,'FINAL MODEL VALUES',8X,
     1'RESIDUAL',8X,'WEIGHTED RESIDUAL'/5X,100(1H*)/)
 1019  FORMAT (I9,4D20.10)
       CALL DYMU(TH,DELZ,NOB,NP)
       NI = -NOB
       DO 1900 I = 1,NP
       NI = NI + NOB
       II = (I-1)*NP
       NJ = - NOB
       IJ = I - NP
       DO 1900 J = 1,I
       NJ = NJ + NOB
       IJ = IJ + NP
       JI = II + J
       SUMD = 0.D0
       DO 1800 K = 1,NOB
 1800  SUMD = SUMD + W(K)*DELZ(NI+K)*DELZ(NJ+K)
       A(IJ) = SUMD
 1900  A(JI) = SUMD
 2000  IDF = NOB - NP
       IF (IO .NE. 0) WRITE( 6,1015)
 1015  FORMAT ('0CORRELATION MATRIX'/)
       DO 2100 I = 1,NPSQ
 2100  D(I) = A(I)
       CALL MATIN (D,NP,A,0,DET,PIVRAT)
       DO 2200 I = 1,NP
       II = (I-1)*NP + I
       IF (D(II) .LT. 1.D-50) D(II) = 1.D-50
 2200  E(I) = DSQRT(D(II))
       NI = - NP
       DO 2400 I = 1,NP
       NI = NI + NP
       IJ = I - NP
       DO 2300 J = 1,I
       JI = NI + J
       IJ = IJ + NP
       D(JI) = D(JI)/(E(I)*E(J))
 2300  D(IJ) = D(JI)
 2400  IF (IO .NE. 0) WRITE (6,2007) I,(D(NI+J),J=1,I)
 2007  FORMAT (5X,'ROW',I5,9F12.6/(13X,9F12.6))
       IF (IO .NE. 0) WRITE (6,31) DET,PIVRAT
 31    FORMAT (/'0DETERMINANT',D20.10,10X,'RATIO OF PIVOTS',D20.10)
       IF (IDF .EQ. 0) GO TO 3000
       SDEV = SSQ/IDF
       IFACTR = STUDNT(IDF)
       IF (IO .NE. 0) WRITE (6,1014) SSQ,SDEV,IDF
 1014  FORMAT (///5X,'SUM OF SQ. RESIDUALS',D20.10/5X,'MEAN SQ. RESIDUALS
     1',D20.10,5X,'(FOR',I8,3X,'DEGREES OF FREEDOM)')
       SDEV = DSQRT(SDEV)
       IF (IO .NE. 0) WRITE (6,2008) SDEV,TFACTR,SUMW
 2008  FORMAT (5X,'STANDARD ERROR OF ESTIMATE',D20.10,10X,'CRITICAL STUDE
     1NT-T VALUE',F10.3/5X,'SUM OF WEIGHTS',D20.10)
       IF (IO .NE. 0) WRITE (6,2009)
 2009  FORMAT (///5X,'NO.',7X,'PARAMETER VALUE',4X,'STANDARD ERROR',7X,
     1'CONFID. BAND HALF-WIDTH'/5X,90(1H*)/)
       DO 2500 I = 1,NP
       TEMP = E(I)*SDEV
```

```
280*            E(I) = TEMP*TFACTR
281*    2500    IF (IO .NE. 0) WRITE (6,2001) I,TH(I),TEMP,E(I)
282*    3000    IF (IO .EQ. 0 .OR. PIVRAT .LT. 1.D-12) RETURN
283*            DO 3100 I = 1,NP
284*            II = (I-1)*NP + I
285*    3100    Q(I) = A(II)
285*            CALL SYMQR (A,NP,NP,P,DELZ)
287*            NI = - NP
288*            DO 3200 I = 1,NP
289*            NI = NI + NP
290*            A(NI + I) = Q(I)
291*            IJ = I - NP
292*            DO 3200 J = 1,I
293*            IJ = IJ + NP
294*    3200    A(IJ) = A(NI+J)
295*            WRITE (6,2003)
296*    2003    FORMAT (//9X,'EIGENVALUES OF MOMENT MATRIX-FINAL ANALYSIS'/)
297*            WRITE (6,2011) (P(I),I=1,NP)
298*            RETURN
299*            END

END OF COMPILATION:          1 DIAGNOSTICS.


        COMPILATION TIME:          4.829 SEC.
        I/O REQUESTS:                124
        I/O WORDS TRANSFERED:      94557
```

E. SYMQR

This program is a reworked version of a Univ. of Wis. Computing Center routine of the same name which computes the eigenvalues and eigenvectors of a N X N symmetric matrix A by means of the Q-R method.

CALL SYMQR (A, N, NDIM, EIG, VEC)

INPUT: A - matrix of which eigenset is to be found. (Only the lower triangle of A is used)

N - size of A

NDIM - dimensioned size of A

OUTPUT: EIG(I) - I-th largest eigenvalue of A

VEC(J,I) - (NDIM X N matrix) - J-th component of eigenvector I .

```
 SYMQR
)0 COMPILED BY FOR-V S7E6 ON 08 SEP 71 AT 10:43:42.

AGE USED: CODE(1) 001561; DATA(0) 001052

 1.          SUBROUTINE SYMQR (A,N,NDIM,EIG,VEC)
 2.    C     USES THE QR ALGORITHM ON SYMMETRIC MATRIX A TO OBTAIN EIGENVALUES
 3.    C     NOTE THAT ONLY LOWER TRIANGLE OF A IS USED (AND DESTROYED)
 4.    C     EIG - VECTOR OF EIGENVALUES IN DESCENDING ORDER
 5.    C     VEC - MATRIX OF EIGENVECTORS IN ORDER OF EIGENVALUES
 6.    C     IND - ERROR RETURN INDICATIR
 7.    C     IF 0 NORMAL, 1 SUM OF EIGENVALUES NOT TRACE
 8.    C     IF 2 SUM OF SQUARED EIG. NOT EQUAL TO NORM
 9.    C     IF 3 BOTH OF ERRORS
10.          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
11.          DIMENSION A(1),GAMMA(50),BETA(50),BETASQ(50),EIG(1),W(49),VEC(1)
12.         1,P(49),Q(49),IPOSV(50),IVPOS(50),IORD(50)
13.          EQUIVALENCE (IPOSV(1),GAMMA(1)),(IVPOS(1),BETA(1)),(IORD(1),BETASQ
14.         1(1)),(P(1),BETA(1)),(Q(1),BETA(1))
15.          IND = 0
16.          IF (N .EQ. 0) GO TO 560
17.          N1 = N - 1
18.          N2 = N - 2
19.          ENORM = 0.D0
20.          TRACE = 0.D0
21.          NJ = - NDIM
22.          DO 110 J = 1,N
23.          NJ = NJ + NDIM
24.          JJ = NJ + J
25.          DO 100 I = J,N
26.          IJ = NJ + I
27.   100    ENORM = ENORM + A(IJ)**2
28.          TRACE = TRACE + A(JJ)
29.   110    ENORM = ENORM - 0.5D0*A(JJ)**2
30.          ENORM = ENORM + ENORM
31.          GAMMA(1) = A(1)
32.          IF (N2) 280,270,120
33.   120    NRN = -NDIM
34.          DO 260 NR = 1,N2
35.          NRN = NRN + NDIM
36.          ISUB = NRN + NR + 1
37.          B = A(ISUB)
38.          S = 0.D0
39.          DO 130 I = NR,N2
40.   130    S = S + A(NRN + I + 2)**2
41.          A(ISUB ) = 0.D0
42.          IF (S .LE. 0.D0) GO TO 250
43.   140    S = S + B*B
44.          SGN  = 1.D0
45.          IF (B .GE. 0.D0) GO TO 160
46.   150    SGN = -1.D0
47.   160    SQRTS = DSQRT(S)
48.          D = 0.5D0*SGN/SQRTS
49.          TEMP = DSQRT(0.5D0 + B*D)
50.          W(NR) = TEMP
51.          A(ISUB ) = TEMP
52.          D = D/TEMP
```

```
53*             B = '-SGN*SQRTS
54*             DO 170 I = NR,N2
55*             TEMP = D*A(NRN + I+ 2)
56*             W(I+1) = TEMP
57*     170     A(NRN + I+ 2) = TEMP
58*             WTAW = 0.DO
59*             DO 220 I = NR,N1
60*             SUM = 0.DO
61*             DO 180 J = NR,I
62*             IJ = J*NDIM + I+ 1
63*     180     SUM = SUM + A(IJ)*W(J)
64*             II = I + 1
65*             IF (N1 .LT. II) GO TO 210
66*     190     DO 200 J = II,N1
67*             IJ = I*NDIM + J + 1
68*     200     SUM = SUM + A(IJ)*W(J)
69*     210     P(I) = SUM
70*     220     WTAW = WTAW + SUM*W(I)
71*             DO 230 I = NR,N1
72*     230     Q(I) = P(I) -WTAW*W(I)
73*             DO 240 J = NR,N1
74*             QJ = Q(J)
75*             WJ = W(J)
76*             DO 240 I = J,N1
77*             IJ = J*NDIM + I+ 1
78*     240     A(IJ) = A(IJ) - 2.DO*(W(I)*QJ + WJ*Q(I))
79*     250     BETA(NR) = B
80*             BETASQ(NR) = B*B
81*     260     GAMMA(NR+1) = A(ISUB + NDIM)
82*     270     IJ = N2*NDIM + N
83*             B = A(IJ)
84*             BETA (N-1) = B
85*             BETASQ(N-1) = B*B
86*             GAMMA(N) = A(IJ + NDIM)
87*     280     BETASQ(N) = 0.DO
88*             DO 300 I = 1,N
89*             IJ = I - NDIM
90*             DO 290 J = 1,N
91*             IJ = IJ + NDIM
92*     290     VEC(IJ) = 0.DO
93*             IJ = (I-1)*NDIM + I
94*     300     VEC(IJ) = 1.DO
95*             M = N
96*             SUM = 0.DO
97*             NPAS = 1
98*             GO TO 400
99*     310     SUM = SUM + SHIFT
100*            COSA = 1.DO
101*            G = GAMMA(1) - SHIFT
102*            PP = G
103*            PPBS = PP*PP + BETASQ(1)
104*            PPBR = DSQRT(PPBS)
105*            NJ = - NDIM
106*            DO 370 J = 1,M
107*            NJ = NJ + NDIM
108*            COSAP = COSA
*DIAGNOSTIC* THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
```

```
      IF (PPBS .NE. 0.D0) GO TO 320
      SINA = 0.D0
      SINA2 = 0.D0
      COSA = 1.D0
      GO TO 350
  320 SINA = BETA(J)/PPBR
      SINA2 = BETASQ(J)/PPBS
      COSA = PP/PPBR
      NT = J + NPAS
      IF (NT .GT. N) NT = N
  330 DO 340 I = 1,NT
      IJ = NJ + I
      TEMP = COSA *VEC(IJ) + SINA*VEC(IJ + NDIM)
      VEC(IJ + NDIM) = -SINA*VEC(IJ) + COSA*VEC(IJ+ NDIM)
  340 VEC(IJ) = TEMP
  350 DIA = GAMMA(J+1) - SHIFT
      U = SINA2 *(G + DIA)
      GAMMA(J) = G + U
      G = DIA - U
      PP = DIA*COSA - SINA*COSAP*BETA(J)
      IF (J .NE. M) GO TO 360
      BETA(J) = SINA*PP
      BETASQ(J) = SINA2*PP*PP
      GO TO 380
  360 PPBS = PP*PP + BETASQ(J+1)
      PPBR = DSQRT(PPBS)
      BETA(J) = SINA*PPBR
  370 BETASQ(J) = SINA2*PPBS
  380 GAMMA(M+1) = G
      NPAS = NPAS + 1
      IF (BETASQ(M) .GT. 1.D - 21) GO TO 410
  390 EIG(M+1) = GAMMA(M+1) + SUM
  400 BETA(M) = 0.D0
      BETASQ(M) = 0.D0
      M = M - 1
      IF (M .EQ. 0) GO TO 430
      IF (BETASQ(M) .LE. 1.D - 21) GO TO 390
  410 A2 = GAMMA(M+1)
      R2 = 0.5D0*A2
      R1 = 0.5D0*GAMMA(M)
      R12 = R1 + R2
      DIF = R1 - R2
      TEMP = DSQRT(DIF*DIF + BETASQ(M))
      R1 = R12 + TEMP
      R2 = R12 - TEMP
      DIF = DABS(A2 - R1) - DABS(A2 - R2)
      IF (DIF .LT. 0.D0) GO TO 420
      SHIFT = R2
      GO TO 310
  420 SHIFT = R1
      GO TO 310
  430 EIG(1) = GAMMA(1) + SUM
      DO 440 J = 1,N
      IPOSV(J) = J
      IVPOS(J) = J
  440 IORD(J) = J
      M = N
```

```
166*            GO TO 470
167*    450     DO 460 J = 1,M
168*            IF (EIG(J) .GE. EIG(J+1)) GO TO 460
169*            TEMP = EIG(J)
170*            EIG(J) = EIG(J+1)
171*            EIG(J+1) = TEMP
172*            ITEMP = IORD(J)
173*            IORD(J) = IORD(J+1)
174*            IORD(J+1) = ITEMP
175*    460     CONTINUE
176*    470     M = M - 1
177*            IF (M .NE. 0) GO TO 450
178*            IF (N1 .EQ. 0) GO TO 500
179*            NL = - NDIM
180*            DO 490 L = 1,N1
181*            NL = NL + NDIM
182*            NV = IORD(L)
183*            NP = IPOSV(NV)
184*            IF (NP .EQ. L) GO TO 490
185*            LV = IVPOS(L)
186*            IVPOS(NP) = LV
187*            IPOSV(LV) = NP
188*            NNP = (NP- 1)*NDIM
189*            DO 480 I = 1,N
190*            IL = NL + I
191*            INP = NNP + I
192*            TEMP = VEC(IL)
193*            VEC(IL) = VEC(INP)
194*    480     VEC(INP) = TEMP
195*    490     CONTINUE
196*    500     ESUM = 0.D0
197*            ESSQ = 0.D0
198*            DO 550 NRR = 1,N
199*            K = N1
200*    510     K = K - 1
201*            IF (K .LE. 0) GO TO 540
202*            SUM = 0.D0
203*            DO 520 I = K,N1
204*            IL = (NRR - 1)*NDIM + I + 1
205*            IJ = (K-1)*NDIM + I + 1
206*    520     SUM = SUM + VEC(IL)*A(IJ)
207*            SUM = SUM + SUM
208*            DO 530 I = K,N1
209*            IL = (NRR - 1)*NDIM + I + 1
210*            IJ = (K-1)*NDIM + I + 1
211*    530     VEC(IL ) = VEC(IL) - SUM*A(I,J)
212*            GO TO 510
213*    540     ESUM = ESUM + EIG(NRR)
214*    550     ESSQ = ESSQ + EIG(NRR)**2
215*            TEMP = DABS(128.D0*TRACE)
*DIAGNOSTIC* THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
216*            IF ((DABS(TRACE - ESUM) + TEMP) - TEMP .NE. 0.D0) IND = IND + 1
217*            TEMP = 256.D0*ENORM
*DIAGNOSTIC* THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
218*            IF ((DABS(ENORM - ESSQ) + TEMP) - TEMP .NE. 0.D0) IND = IND + 2
219*            IF (IND .NE. 0) WRITE (6,86) IND
220*    86      FORMAT ('0** PROBABLE ERROR IN SYMQR** INDICATOR=',I5)
```

```
221*    560    RETURN
222*           END

END OF COMPILATION:          3 DIAGNOSTICS.

        COMPILATION TIME:          3.176 SEC.
        I/O REQUESTS:                 92
        I/O WORDS TRANSFERED:      79676
SYMQR
018A-09/08-10143
```

F. MATIN

Program which solves the set of equations $AX = B$, and inverts A, by means of Gaussian elimination with no pivoting.

CALL MATIN (A, N, B, NB, DET, PIVRAT)

where:

A - N X N matrix of equations (replaced by A^{-1})

N - size of A

B - N X NB matrix of right-hand side of equations (replaced by solution matrix X)

NB - no. of different right-hand sides (if 0, B is ignored by routine)

DET - returned determinant of A

PIVRAT - set to ratio of minimum to maximum pivots ($-\log_{10}$PIVRAT is approx. no. of decimal figures lost in the computations).

```
FOR[illegible]I MATIN
YCLE 000 COMPILED BY FOR-V S601 ON 08 AUG 70 AT 18:34:36.

0101      1*            SUBROUTINE MATIN (A,N,B,NB,DET,PIVRAT)
0103      2*            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0104      3*            DIMENSION A(1),B(1)
0105      4*            AL = DABS(A(1))
0106      5*            AU = AL
0107      6*            DET = 1.DO
0110      7*            NK = - N
0111      8*            DO 1000 K = 1,N
0114      9*            NK = NK + N
0115     10*            KK = NK + K
0116     11*            PIVOT = A(KK)
0117     12*            PIVRAT = DABS(PIVOT)
0120     13*            IF (AL .GT. PIVRAT) AL = PIVRAT
0122     14*            IF (AU .LT. PIVRAT) AU = PIVRAT
0124     15*            DET = DET*PIVOT
0125     16*            IF (PIVRAT .LT. 1.D-50) PIVOT = 1.D-50
0127     17*            A(KK) = 1.DO
0130     18*            PIVOT = 1.DO/PIVOT
0131     19*            KI = K - N
0132     20*            DO 100 I = 1,N
0135     21*            KI = KI + N
0136     22*      100   A(KI) = A(KI) *PIVOT
0[illegible]     23*            IF (NB .EQ. 0) GO TO 300
0142     24*            KI = K - N
0143     25*            DO 200 I = 1,NB
0146     26*            KI = KI + N
0147     27*      200   B(KI) = B(KI) *PIVOT
0151     28*      300   DO 1000 J = 1,N
0154     29*            IF (J .EQ. K) GO TO 1000
0156     30*            JK = NK + J
0157     31*            T = A(JK)
0160     32*            A(JK) = 0.DO
0161     33*            JI = J - N
0162     34*            KI = K - N
0163     35*            DO 400 I = 1,N
0166     36*            JI = JI + N
0167     37*            KI = KI + N
0170     38*      400   A(JI) = A(JI) - A(KI)*T
0172     39*            IF (NB .EQ. 0) GO TO 1000
0174     40*            JI = J - N
0175     41*            KI = K - N
0176     42*            DO 500 I = 1,NB
0201     43*            JI = JI + N
0202     44*            KI = KI + N
0203     45*      500   B(JI) = B(JI) - B(KI)*T
0205     46*     1000   CONTINUE
0210     47*            PIVRAT = AL/AU
0[illegible]     48*            RETURN
0[illegible]     49*            END

      END OF COMPILATION:          NO  DIAGNOSTICS.
```

G. STUDNT

This function returns the value of the student-t statistic for IDF degrees of freedom at the 95% level. (Equal-Tails).

```
T = STUDNT (IDF)
```

```
;I STJDNT
 000 COMPILED BY FOR-V S7E6 ON 08 SEP 71 AT 10:43:59.

ORAGE USED: CODE(1) 000070; DATA(0) 000052

     1*            FUNCTION STUDNT (IDF)
     2*      C     RETURNS 95 PERCENT STUDENT-T VALUE FOR IDF DEG. OF FREEDOM
     3*            DIMENSION T(29)
     4*            DATA T/12.706,4.303,3.182,2.776,2.571,2.447,2.365,2.306,2.262,2.22
     5*           18,2.201,2.179,2.160,2.145,2.131,2.120,2.110,2.101,2.093,2.086,2.08
     6*           20,2.074,2.069,2.064,2.060,2.056,2.052,2.048,2.045/
     7*            IF (IDF .LT. 60) GO TO 100
     8*            STUDNT = 1.960 + 2.4/IDF
     9*      50    RETURN
    10*      100   IF (IDF .LT. 30) GO TO 200
    11*            STUDNT = 1.958 + 2.52/IDF
    12*            GO TO 50
    13*      200   IF (IDF .LE. 0) GO TO 300
    14*            STUDNT = T(IDF)
    15*            GO TO 50
    16*      300   STUDNT = 1.E+30
    17*            GO TO 50
    18*            END

 END OF COMPILATION:        NO  DIAGNOSTICS.


          COMPILATION TIME:          0.282 SEC.
          I/O REQUESTS:                46
          I/O WORDS TRANSFERED:      56744
```

H. EXAMPLES

The example is 48 cross sections Q(v) for $1 < E^* < 40$ generated by means of XSECT (see Sect. II) from a L-J (12,6) potential with $\varepsilon = 13.3$ cpe, $r_m = 4.92$ Å (intended to mimic Li-Hg), $\mu = 9.69925$ ppg., $g_o = 0.186299$, $a_1 = 0.421559$, $C_6 = 0.378 \times 10^6$ cpe-Å^6. ($\sigma = 0.25\%$)

1. True value of $\gamma_5 = 0$

(a) NP = 5 (No C_8)

(1) Note that the correlation between $\gamma_1(C_6)$ and the other parameters is small.

(2) The "STANDARD ERROR OF ESTIMATE" is 0.262%, which is compatible with the precision of the calculated Q(v)'s.

(b) NP = 6 (C_8 assumed present)

(1) STANDARD ERROR OF ESTIMATE IS 0.259%, no change from NP = 5, indicating γ_6 is not necessary to the fit.

(2) The correlation between γ_1 and γ_6 is high. (-0.98)

(3) The determined value of C_8 is not significantly different from zero.

(4) The precision of C_6 has decreased markedly.

2. True value of $\gamma_5 = 2$

(a) NP = 5 (C_8 assumed zero)

(1) STANDARD ERROR OF ESTIMATE = 0.388% (larger than before)

(2) Precision of C_6 is lower than before.

(b) NP = 6 (C_8 assumed present)

(1) STANDARD ERROR OF ESTIMATE = 0.260%, lowered from NP = 5.

(2) C_8 is found to be significantly greater than zero.

(3) C_6 still has comparatively low precision.

It is to be noted that the presence of C_8, either real or imagined, results in an uncertainty in the value of C_6.

TEST RUN ON O'BRIEN Q'S -- FAST CROSS SECTION GENERATOR

NO OF DATA POINTS 48
REDUCED MASS .969925+01
INITIAL EPS .140000+02
INITIAL RM .470000+01
INITIAL A1 .440000+00
INITIAL G0 .190000+00
INITIAL C6 .378000+06
WEIGHTING SWITCH 1

FIT WITH 5 PARAMETERS (NO C8)

(2)

ENTRY TO GASS59

FIT TO 48 OBSERVATIONS
THERE ARE 5 PARAMETERS IN THE MODEL

INITIAL PARAMETER VALUES

	LOWER BOUND	PARAMETER VALUE	UPPER BOUND
1	.3000000000+000	.1346900506+004	.1000000000+009
2	.0000000000+000	.8368514457+001	.1000000000+003
3	.0000000000+000	.1400000000+002	.5000000000+002
4	.0000000000+000	.1093130943+003	.2000000000+005
5	.0000000000+000	.3494053018+004	.1000000000+005

INITIAL SUM OF SQUARES .6924430832-001

ITERATION STOPS ... RELATIVE CHANGE IN EACH PARAMETER IS LESS THAN .1000000000-002

CORRELATION MATRIX

ROW	1	1.000000				
ROW	2	-.125735	1.000000			
ROW	3	.119972	-.785788	1.000000		
ROW	4	.069570	-.619235	.774383	1.000000	
ROW	5	.101764	-.763495	.964803	.903321	1.000000

DETERMINANT .4639324922-022 RATIO OF PIVOTS .1051003265-005

SUM OF SQ. RESIDUALS .2956066462-003
MEAN SQ. RESIDUALS .6874573167-005 (FOR 43 DEGREES OF FREEDOM)
STANDARD ERROR OF ESTIMATE .2621940725-002 CRITICAL STUDENT-T VALUE 2.017
SUM OF WEIGHTS .7377959458-004

NO.	PARAMETER VALUE	STANDARD ERROR	CONFID. BAND HALF-WIDTH
1	.1346493596+004	.5161492449+000	.1040863958+001
2	.8849899637+001	.1875344027+000	.3781827452+000
3	.1399492168+002	.5074537144+000	.1023333511+001
4	.1047705400+003	.1596391954+000	.3219291409+000
5	.3369161074+004	.6615348134+002	.1334356560+003

EIGENVALUES OF MOMENT MATRIX-FINAL ANALYSIS

.6864768959-002 .5771285384-003 .2862986311-003 .2605084319-004 .1570056419-00

TERM	VALUE	95 PERCENT HALFWIDTH
C6	.377715+06	.729954+03
G0	.211935+00	.627958-01
EPS	.139949+02	.102333+01
A1	.435199+00	.363346-01
RM	.453529+01	.687149+00
A1*EPS*RM	.276225+02	.848760-01
RM*EPS**2	.888272+03	.351800+02

FIT WITH 6 PARAMETERS (WITH C8)

ENTRY TO GASS59

FIT TO 48 OBSERVATIONS
THERE ARE 6 PARAMETERS IN THE MODEL

INITIAL PARAMETER VALUES

	LOWER BOUND	PARAMETER VALUE	UPPER BOUND
1	.0000000000+000	.1346493596+004	.1000000000+009
2	.0000000000+000	.8849899637+001	.1000000000+003
3	.0000000000+000	.1399492168+002	.5000000000+002
4	.0000000000+000	.1047705400+003	.2000000000+005
5	.0000000000+000	.3359161074+004	.1000000000+005
6	.0000000000+000	.0000000000+000	.1000000000+009

INITIAL SUM OF SQUARES .2956066462-003

ITERATION STOPS ... RELATIVE CHANGE IN EACH PARAMETER IS LESS THAN .1000000000-002

CORRELATION MATRIX

ROW 1	1.000000					
ROW 2	.109323	1.000000				
ROW 3	-.067173	-.787775	1.000000			
ROW 4	-.112322	-.620076	.776645	1.000000		
ROW 5	-.096477	-.767335	.964894	.904750	1.000000	
ROW 6	-.980106	-.136736	.092670	.128555	.118859	1.000000

DETERMINANT .1298484915-027 RATIO OF PIVOTS .1049823447-005

SUM OF SQ. RESIDUALS .2827158327-003
MEAN SQ. RESIDUALS .6731329349-005 (FOR 42 DEGREES OF FREEDOM)
STANDARD ERROR OF ESTIMATE .2594480555-002 CRITICAL STUDENT-T VALUE 2.013
SUM OF WEIGHTS .7377959458-004

NO.	PARAMETER VALUE	STANDARD ERROR	CONFID. BAND HALF-WIDTH
1	.1343020206+004	.2573184744+001	.5192686754+001
2	.9835969663+001	.1873918117+000	.3781566717+000
3	.1398635876+002	.5053793366+000	.1019855490+001
4	.1047772472+003	.1596028352+000	.3220785178+000
5	.3370098131+004	.6609026839+002	.1333701601+003
6	.2118036661+001	.1541351037+001	.3110445357+001

5

EIGENVALUES OF MOMENT MATRIX-FINAL ANALYSIS

.6837116884-002 .5765694685-003 .2864523427-003 .9792655089-004 .7620278987-0

TERM	VALUE	95 PERCENT HALFWIDTH
C6	.375283+06	.362751+04
G0	.211061+00	.720543-01
EPS	.139864+02	.101986+01
A1	.434840+00	.488882-01
RM	.454211+01	.839337+00
A1*EPS*RM	.276243+02	.849153-01
RM*EPS**2	.888520+03	.351628+02
BETA	.101129+01	.148513+01
C8	.379520+06	.561013+06

TEST RUN ON O'BRIEN O'S --- WITH C8 = C6

NO OF DATA POINTS	48
REDUCED MASS	.969925+01
INITIAL EPS	.140000+02
INITIAL RM	.470000+01
INITIAL A1	.440000+00
INITIAL G0	.190000+00
INITIAL C6	.378000+06
WEIGHTING SWITCH	1

FIT WITH 5 PARAMETERS (NO C8)

ENTRY TO GASS59

FIT TO 48 OBSERVATIONS
THERE ARE 5 PARAMETERS IN THE MODEL

INITIAL PARAMETER VALUES

	LOWER BOUND	PARAMETER VALUE	UPPER BOUND
1	.0000000000+000	.1346900506+004	.1000000000+009
2	.0000000000+000	.8368514457+001	.1000000000+003
3	.0000000000+000	.1400000000+002	.5000000000+002
4	.0000000000+000	.1098130943+003	.2000000000+005
5	.0000000000+000	.3494053018+004	.1000000000+005

INITIAL SUM OF SQUARES .6944308972-001

ITERATION STEPS ... RELATIVE CHANGE IN SUM OF SQUARES IS LESS THAN .1000000000-002

CORRELATION MATRIX

ROW	1	1.000000				
ROW	2	-.125849	1.000000			
ROW	3	.120120	-.785519	1.000000		
ROW	4	.069694	-.619106	.774527	1.000000	
ROW	5	.101897	-.763207	.964780	.903471	1.000000

DETERMINANT .4538018739-022 RATIO OF PIVOTS .1048084199-005

SUM OF SQ. RESIDUALS .6485054621-003
MEAN SQ. RESIDUALS .1508152237-004 (FOR 43 DEGREES OF FREEDOM)
STANDARD ERROR OF ESTIMATE .3883493578-002 CRITICAL STUDENT-T VALUE 2.017
SUM OF WEIGHTS .7339488055-004

NO.	PARAMETER VALUE	STANDARD ERROR	CONFID. BAND HALF-WIDTH
1	.1346277770+004	.7663563823+000	.1545437330+001
2	.8842247107+001	.2781089456+000	.5608357880+000
3	.1401673648+002	.7529242141+000	.1518350458+001
4	.1047770373+003	.2372077541+000	.4783542557+000
5	.3372315152+004	.9828877413+002	.1982095972+003

EIGENVALUES OF MOMENT MATRIX-FINAL ANALYSIS

.6840167958-002 .5749404385-003 .2851315924-003 .2592519684-004 .1561022854-0

TERM	VALUE	95 PERCENT HALFWIDTH
C6	.377563+06	.108354+04
G0	.212611+00	.786564-01
EPS	.140167+02	.151835+01
A1	.435497+00	.219404-01
RM	.452542+01	.727141+00
A1*EPS*RM	.276243+02	.126117+00
RM*EPS**2	.889104+03	.522576+02

FIT WITH 6 PARAMETERS (WITH C8)

ENTRY TO GASS59

FIT TO 48 OBSERVATIONS
THERE ARE 6 PARAMETERS IN THE MODEL

INITIAL PARAMETER VALUES

	LOWER BOUND	PARAMETER VALUE	UPPER BOUND
1	.0000000000+000	.1346277770+004	.1000000000+009
2	.0000000000+000	.8842247107+001	.1000000000+003
3	.0000000000+000	.1401673648+002	.5000000000+002
4	.0000000000+000	.1047770373+003	.2000000000+005
5	.0000000000+000	.3372315152+004	.1000000000+005
6	.0000000000+000	.0000000000+000	.1000000000+009

INITIAL SUM OF SQUARES .6485054621-003

ITERATION STOPS ... RELATIVE CHANGE IN SUM OF SQUARES IS LESS THAN .1000000000-002

CORRELATION MATRIX

ROW 1	1.000000					
ROW 2	.108430	1.000000				
ROW 3	-.066259	-.787424	1.000000			
ROW 4	-.111588	-.625830	.776739	1.000000		
ROW 5	-.895596	-.766952	.964858	.904885	1.000000	
ROW 6	-.980109	-.135852	.091774	.127835	.117993	1.000000

DETERMINANT .1264424246-027 RATIO OF PIVOTS .1046818301-003

SUM OF SQ. RESIDUALS .2834185229-003
MEAN SQ. RESIDUALS .6748060070-005 (FOR 42 DEGREES OF FREEDOM)
STANDARD ERROR OF ESTIMATE .2597702845-002 CRITICAL STUDENT-T VALUE 2.019
SUM OF WEIGHTS .7339488055-004

NO.	PARAMETER VALUE	STANDARD ERROR	CONFID. BAND HALF-WIDTH
1	.1342982568+004	.2582873877+001	.5212239424+001
2	.8828489430+001	.1878008742+000	.3789821599+000
3	.1401071320+002	.5067464533+000	.1022514331+001
4	.1047842052+003	.1602860629+000	.3234572713+000
5	.3373520804+004	.6634987007+002	.1338940363+003
6	.4127919393+001	.1547427233+001	.3122708120+001

EIGENVALUES OF MOMENT MATRIX-FINAL ANALYSIS

10

.6814585221-002 .5744860009-003 .2853441146-003 .9742674141-004 .7579928003-006

TERM	VALUE	95 PERCENT HALFWIDTH
C6	.375257+06	.36402+04
G0	.211848+00	.724291-01
EPS	.140107+02	.102261+01
A1	.435184+00	.490069-01
RM	.453093+01	.838421+00
A1*EPS*RM	.276261+02	.852789-01
RM*EPS**2	.889422+03	.353009+02
BETA	.197094+01	.149098+01
C8	.739608+06	.566678+06

VII. JWKB

This main program computes JWKB phase shifts $\eta_\ell(E)$ for given reduced energies $E^* = E/\epsilon$ and values of the orbital quantum number ℓ such that $b^* = (\ell+\frac{1}{2})/ka$ is a multiple of 0.05 (For details of the calculation of η_ℓ, see function PHASHT).

The total cross section is given by

$$Q(E) = \frac{4\pi}{k^2} \sum_{\ell=0}^{\infty} (2\ell+1) \sin^2 \eta_\ell \qquad \text{(VII-1)}$$

Letting

$$\delta q_\ell = b \sin^2 \eta_\ell \qquad \text{(VII-2)}$$

and

$$q_L = \sum_{\ell=0}^{L} \delta q_\ell \qquad \text{(VII-3)}$$

Since $Q = \frac{8\pi}{k} q_\infty$, the program computes Q by means of

$$Q = \frac{8\pi}{k} (q_L + \Delta q_L) \qquad \text{(VII-4)}$$

where L is a value large enough so that $\Delta q_L = q_\infty - q_L$ can be estimated accurately.

Values of δq_ℓ are calculated at every 0.05 in b^* until one of the following is true:

(1) $b^* > 2$ and $\delta Q = \frac{8\pi}{k} \delta q_\ell < 0.1$ Å^2/b (This condition forces the remaining cross section ΔQ to be less than 0.05 Å^2).

(2) The Jeffreys-Born long-range phase shift $\eta_\ell^{JB} = \frac{3\pi}{16}\frac{C_6}{\hbar v b} < 2$ and agrees with η_ℓ to within 0.01 radians for 3 successive values of ℓ .

When either condition is satisfied, this value of ℓ is taken as L .

The intermediate values of η_ℓ are then calculated by interpolation (DNTERP)*. The remaining cross section $\Delta Q = \frac{8\pi}{k}\Delta q_L$ is

---

* The program could be improved by replacement of DNTERP with a continued-fraction interpolation program, since the limiting form of η_ℓ is $\propto 1/b$.

---

estimated two different ways:

$$(1)\quad \Delta q_L = \frac{\eta_L}{\eta_L^{JB}} \int_L^\infty b \sin^2 \eta_\ell^{JB}\, d\ell \quad \text{(QJBVAL)} \qquad \text{(VII-5)}$$

(The factor of η_L/η_L^{JB} is to correct for the discrepancy between η_ℓ and η_ℓ^{JB})

(2) $q_\infty = \lim_{\ell\to\infty} q_\ell$ is obtained by means of the ε-algorithm (EPSALG).

Finally, the apparent C_6 constant C_6^{app} is calculated by means of:

$$C_6^{app} = \frac{\hbar^2 k}{\mu}\left(\frac{Q}{8.083}\right)^{5/2} \qquad \text{(VII-6)}$$

INPUT:

Card 1: Title (20A4)

Card 2: NE, MU (I5, F10.4)

Card 3: (ELIST(I), I = 1,NE) (10F8.0)

where:

NE - No. of energies (in reduced units) for which cross sections are to be calculated

MU - reduced mass of system in ppg.

ELIST(I) - I-th energy in reduced units

NOTE: In addition, a user defined function POTIN/POT is required (q.v.)

OUTPUT: The values of E, C_6, and σ (zero of potential) are in reduced units. The quantities EUNIT = e and RUNIT = a are the reducing units of energy and distance returned by POTIN (q.v.): they should be of the order of ε and r_m, respectively.

For ℓ corresponding to every 0.05 in $b^* = (\ell + \frac{1}{2})/A$, the following are outputted:

L - ℓ

B - b^*

YO - $y_o = 1/r_o$

RO - r_o the outer most zero of $1-b^2/r^2 - V(r)/E$ (in reduced units)

ETATAB(LL) - η_ℓ

ETAJB - η_ℓ^{JB}

DELQ - δq_ℓ (in reduced units)

IER - Error return from determination of turning point. If = 0, normal convergence. If $\neq$ 0 turning point is incorrect (see REGFAL)

ISW - Error return from determination of η_ℓ. If = 0, normal convergence. If $\neq$ 0, did not converge (see PHASHT).

A table of the above quantities is printed out, the units being those obtained treating EUNIT and RUNIT as dimensional units.

Finally, the ε-algorithm sequence of diagonal convergents (EPSVEC(I), I = 1,M) are printed out, M being the number of applications of the algorithm, and the convergents for q_∞ appearing at every other element (the first such appearance being the one used). The following are then printed out:

T - Execution time in seconds

CROSS - $Q_L = \frac{8\pi}{k} q_L$ $(\mathring{A}^2)$

Q - q_L

QJB - $Q_L + \Delta Q^{JB}$ from eq.(VII-5) $(\mathring{A}^2)$

C6APP - $C_6^{app,JB}$ $(cpe\text{-}\mathring{A}^6)$

ND - highest degree used in interpolation (see DIFTAB)

and finally,

CROSS - Q = $\frac{8\pi}{k} q_\infty$

EPSVEC(J) - q_∞

C6APP - C_6^{app}

NOTE: Values of q outputted are in reduced units.

REMARKS: (1) A user-defined potential function POTIN/POT is required (q.v.)

(2) EUNIT and RUNIT are treated as dimensional units except for actual cross sections which are in $\mathring{A}^2$.

(3) The subroutines TIMSET (in line 54) and TIMGET (in line 93) are specific to the UWCC Univac 1108, and set and read an internal clock. These functions are not necessary for execution of the program. Line 54 may be deleted and line 93 replaced with "600 CONTINUE"

(4) The convergence tolerances for the calculation of η_ℓ, convergence to η_ℓ^{JB}, and determination of the turning point, are given by EPS (absolute), TOLJB (relative/absolute), TOLYO (relative), in line 8.

(5) The phase shift calculations generally terminate at $b^* < 3$ and give a cross section which has limits of error $\pm 5\text{Å}^2$.

```
JWKB
=05/31/72=01:53:12 (,0)      JWKB
    1.      C       PROGRAM WHICH COMPUTES JWKB PHASE SHIFTS AND TOTAL CROSS SECTIONS
    2.      C       A CONSISTENT SET OF UNITS FOR THIS PROGRAM ARE
    3.      C       ANGSTROMS, PICOPICOGRAMS,AND DECIPICOSECONDS (GIVING CENTIPICOERGS
    4.      C       AS THE UNIT OF ENERGY)
    5.              REAL MU
    6.              DIMENSION ELIST(200),ETATAB(200),DIF(2000),EPSVEC(20)
    7.              COMMON B,RO,E,X
    8.              DATA EPS/0.005/,TOLJB/0.01/,TOLYO/1.E-5/
    9.              DATA  PI2/1.57079633/,HX2/1.11216241/
   10.              EXTERNAL VEFF
   11.              WRITE (6,2) TOLYO,EPS,TOLJB
   12.      2       FORMAT (1H1,5X,'ENTRY TO JWKB/DNTERP '////5X,'TOLERANCE FOR YO (RE
   13.             1LATIVE)',E12.5/5X,'TOLERANCE FOR ETA (ABS)',E12.5/5X,'TOLERANCE FO
   14.             2R JB LIMIT',E12.5)
   15.      C       READ IN TITLE OF EXPERIMENT
   16.              READ (5,3) (DIF(I),I=1,20)
   17.      3       FORMAT (20A4)
   18.              WRITE (6,5) (DIF(I),I=1,20)
   19.      5       FORMAT (1H1,10X,20A4)
   20.      C       READ IN NO.  OF ENERGIES NE AND REDUCED MASS MU
   21.              READ (5,6) NE,MU
   22.      6       FORMAT (I5,F10.4)
   23.      C       READ IN  LIST OF ENERGIES
   24.              READ (5,7) (ELIST(I),I=1,NE)
   25.      7       FORMAT (10F8.0)
   26.      C       POTIN SHOULD RETURN THE VALUE OF THE ZERO OF THE POTENTIAL
   27.      C       C6 SHOULD BE VALUE OF LONGE-RANGE C6 IN REDUCED UNITS, THE
   28.      C       SCALE OF WHICH ARE GIVEN IN EUNIT(ENERGY) AND RUNIT (LENGTH)
   29.              SIGMA = POTIN(C6,EUNIT,RUNIT)
   30.              SIGINV = 1./SIGMA
   31.              SIGIN2 = 2.*SIGINV
   32.              CONSJB = 0.1875*PI2*C6
   33.              ECONS = 2.*MU*EUNIT/HX2
   34.              WRITE (6,9) MU,C6,SIGMA,EUNIT,RUNIT
   35.      9       FORMAT ('///5X,'PHASE SHIFTS AND CROSS SECTIONS FOR REDUCED MASS',
   36.             1 E14.6/5X,'LONG-RANGE C6',E14.6/5X,'ZERO OF POTENTIAL',E14.6/
   37.             2 5X,'UNITS OF ENERGY AND LENGTH',2E14.6)
   38.              DO 1000 I = 1,NE
   39.              E = ELIST(I)
   40.              WAVE = SQRT(E*ECONS)
   41.              A = WAVE*RUNIT
   42.              CJB = CONSJB*A/E
   43.              COEF = 16.*PI2*RUNIT/WAVE
   44.              WRITE (6,10) E,WAVE,A
   45.      10      FORMAT ('//5X,'ENERGY',F10.5,5X,'WAVE NO.',E15.6,5X,'A=WAVE*RUNIT'
   46.             1,E14.6//5X,'FOLLOWING TABLE IN REDUCED UNITS EUNIT,RUNIT'/)
   47.              WRITE (6,15)
   48.      15      FORMAT (14X,'L',5X,'B',8X,'YO',10X,'RO',9X,'ETA',11X,'ETAJB',
   49.             1 9X,'DELQ',7X,'IER',3X,'ISW'/5X,100(1H*)/)
   50.              IJB = 0
   51.      C       CALCULATE PHASE SHIFTS AT EVERY .05 IN REDUCED IMPACT PARAMETER
   52.              LSTEP = A/20.
   53.              IF (LSTEP .LT. 1 ) LSTEP = 1
   54.              CALL TIMSET(0.)
   55.              DO 500 LL = 1,200
```

```
      NETA = LL
      L = LSTEP*(LL-1)
      B = (L+0.5)/A
      IF (B .GT. SIGMA) GO TO 100
      XL = SIGINV
      XU = SIGIN2
      GO TO 200
  100 XL = 1./B
      XU = SIGINV
C     FIND TURNING POINT YO = 1/RO
C     IER = 0, NORMAL. IF NOT 0, ERROR IN ROOT (SEE REGFAL)
  200 CALL REGFAL(XL,XU,X,VEFF,TOLYO,5,20,IER)
      RO = 1./X
C     CALCULATE PHASE SHIFT ETA
C     ISW = 0, NORMAL. IF NOT 0, THEN DID NOT CONVERGE TO EPS ACCURACY
      TOL = EPS/A
      ETAJB = CJB/B**5
      IF (ETAJB .LT. 1.) TOL = ETAJB*TOL
      ETATAB(LL) = A*PHASHT(TOL,9,ISW)
      DELQ = B*(SIN(ETATAB(LL)))**2
      WRITE (6,20) L,B,X,RO,ETATAB(LL),ETAJB,DELQ,IER,ISW
   20 FORMAT (10X,I5,3F10.6,3E14.6,2I6)
      IF (COEF*B*DELQ .GT. 0.1 .OR. B .LT. 2.) GO TO 400
      WRITE (6,22)
   22 FORMAT (/5X,'REMAINING CROSS SECTION IS LESS THAN 0.05 SQ ANG')
      GO TO 600
  400 TOL = TOLJB
      IF (ETAJB .LT. 1.) TOL = ETAJB*TOL
      IF (ABS(ETATAB(LL) - ETAJB) .LT. TOL  ) IJB = IJB + 1
      IF (IJB .LT. 3 .OR. ETAJB .GT. 2.) GO TO 500
      WRITE (6,23) IJB
   23 FORMAT(/5X,'AFTER ',I8,5X,'ASYMPTOTIC PHASE SHIFTS, ETAJB IS LESS
     1THAN 1')
      GO TO 600
  500 CONTINUE
      WRITE (6,25)
   25 FORMAT (10X,'MORE THAN  200 PARTIAL WAVES ARE REQUIRED')
  600 T = TIMGET(0)
      ND = 10
      CALL DIFTAB(ETATAB,NETA,DIF,ND,EPS)
      NMAX = LSTEP*(NETA-1) + 1
      BSTEP = LSTEP/A
      BO = 0.5/A
      M = 0
      IEPS = 0
      Q = 0.
      DO 700 LL = 1,NMAX
      L = LL - 1
      B = (L+0.5)/A
      TOL = EPS
      ETAJB = CJB/B**5
      IF (ETAJB .LT. 1.) TOL = ETAJB*TOL
      ETA = DNTERP(B,BO,BSTEP,ETATAB,NETA,DIF,ND,TOL)
      DELQ = B*(SIN(ETA))**2
      Q = Q + DELQ
      IF (LL .LE. NMAX - 20 .OR. IEPS .EQ. 1) GO TO 700
      CALL EPSALG (Q,M,EPSVEC)
```

```
113.            J = 2 + 2*(M/2) - M
114.            IF (M .LE. 2) GO TO 700
115.            IF (ABS(EPSVEC(J) - EPSVEC(J+2)) *COEF .GT. 0.1) GO TO 700
116.            IEPS = 1
117.            WRITE (6,21) (EPSVEC(K),K=1,M)
118.      21    FORMAT (/5X,'EPSILON ALGORITHM CONVERGES ...'/(5X,10F11.4))
119.            GO TO 800
120.      700   CONTINUE
121.      800   B = B + 1./A
122.            QJB = Q + A*ETA*QJBVAL(CJB,B)/ETAJB
123.            CROSJB = COEF*QJB
124.            CROSS = COEF*Q
125.            C6APP = HX2*WAVE*(CROSJB/8.083)**2.5/MU
126.            WRITE (6,30) T,CROSS,Q,CROSJB,QJB,C6APP,ND
127.      30    FORMAT (//5X,'TIME (SEC)',F10.3/10X,'CROSS SECTION AT LAST PHASE S
128.           1HIFT',F10.2,10X,'Q',F10.3/10X,'JB EXTRAPOLATED CROSS SECTION (SQ A
129.           2NG)',F10.2,10X,'QJB',F10.3,10X,'C6 APPARENT',E15.6/10X,'HIGHEST OR
130.           3DER DIFFERENCE USED IN INTERPOLATION',I8)
131.            IF (M .EQ. 0) GO TO 1000
132.            CROSS = COEF*EPSVEC(J)
133.            C6APP = HX2*WAVE*(CROSS/8.083)**2.5/MU
134.            WRITE (6,40) CROSS,EPSVEC(J),C6APP
135.      40    FORMAT (10X,'EPSILON ALGORITHM CROSS SECTION',F10.2,10X,'Q',F10.3,
136.           1 10X,'C6 APPARENT',E14.6)
137.      1000  CONTINUE
138.            STOP
139.            END

ND OF COMPILATION:        NO  DIAGNOSTICS.
```

A. PHASHT

This function computes the value of the JWKB phase shift η_ℓ for given E and $b = (\ell + \frac{1}{2})/k$, to a given absolute tolerance, by means of the formulae

$$\eta_\ell = \eta_1 + \eta_2 \qquad (b > r_0)$$

$$\eta_\ell = \frac{\pi}{2} b - r_0 - \int_0^{y_0} \frac{1-F^{1/2}}{y^2}\, dy \qquad (b \le r_0) \tag{VII-7}$$

where V(r) is the potential, r_0 is the largest positive zero of F(r)

$$F(r) = 1 - \frac{b^2}{r^2} - \frac{V(r)}{E}$$

and $y = 1/r$ ($y_0 = 1/r_0$). The quantities η_1 and η_2 are given by

$$\eta_1 = \int_{1/b}^{y_0} \frac{F^{1/2}}{y^2}\, dy$$

$$\eta_2 = \int_0^{1/b} \frac{F^{1/2} - \sqrt{1-b^2y^2}}{y^2}\, dy \tag{VII-8}$$

The integrals are performed by function ADSIMP (q.v.)

CALL: ETA = PHASHT (EPS, NITER, ISW)

INPUT: EPS - Absolute error tolerance for integrals in Eq. (VII-7,8)

NITER - 2^{NITER} is the maximum number of points which may be used in the integrations.

OUTPUT: ISW - = 0, Normal convergence of integrations; if ≠ 0, integrals did not converge (see ADSIMP)

PHASHT - value of η_ℓ (b,E)

NOTE: Values of B(b), RO(r_o), and YO(y_o) are obtained through the COMMON statement of line 10.

```
M PHASHT
9=05/31/72=01153117 (,0)      PHASHT
    1.              FUNCTION  PHASHT (EPS,NITER,ISW)
    2.        C     FINDS VALUE OF JWKB PHASE SHIFT FOR IMPACT PARAMETER B, TURNING
    3.        C     POINT RO,  EPS IS ABS ERROR CRITERION. 2**NITER IS MAX NO. OF
    4.        C     POINTS WHICH MAY BE USED IN INTEGRATION (NITER .LE. 9)
    5.        C     IF ISW .NE. 0, FAILED TO CONVERGE TO ACCURACY EPS
    6.        C     NOTE = REQUIRES FUNCTIONS F,F2,F3, WHICH FOR Y = 1/R RETURN
    7.        C        F(Y) = (1-SQRT(1-(B*Y)**2 - V(R)/E))/(Y*Y)
    8.        C        F2(Y) = SQRT(1-(B*Y)**2-V(R)/E)/(Y*Y)
    9.        C        F3(Y) = (SQRT(1-(B*Y)**2-V(R)/E)-SQRT(1-(B*Y)**2))/(Y*Y)
   10.              COMMON B,RO,E,YO
   11.              DATA PI2/1.57079633/
   12.              EXTERNAL F,F2,F3
   13.              LIMIT  = 2**NITER
   14.              IF (B .LE. RO) GO TO 100
   15.              BINV = 1./B
   16.              TOL = 0.5*EPS
   17.              ETA1 = ADSIMP(BINV,YO,F2,TOL,LIMIT,ISW1)
   18.              ETA2 = ADSIMP(0.,BINV,F3,TOL,LIMIT,ISW2)
   19.              ISW = ISW1 + ISW2
   20.              PHASHT = ETA1 + ETA2
   21.              RETURN
   22.        100   PHASHT = PI2*B - RO*ADSIMP(0.,YO,F,EPS,LIMIT,ISW)
   23.              RETURN
   24.              END

END OF COMPILATION:          NO  DIAGNOSTICS.
```

B. VEFF/F/F2/F3

Four entry-point function which returns values of

$VEFF(y) = 1 - b^2y^2 - V(r)/E$

$F(y) = (1 - \sqrt{VEFF})/y^2$

$F2(y) = \sqrt{VEFF}\ /y^2$

$F3(y) = (\sqrt{VEFF}\ - \sqrt{1-b^2y^2}\ )/y^2$

where $y = 1/r$, and b and E are obtained via the COMMON statement of line 2.

CALLS Z = VEFF(X)

Z = F(X)

Z = F2(X)

Z = F3(X)

ARGUMENT: X = y

```
F
?05/31/72-01:53:15 (,0)      F
    1.              FUNCTION VEFF(X)
    2.              COMMON B,RO,E,YO
    3.              U = B*X
    4.              VEFF = 1. - U*U - POT(1./X)/E
    5.              RETURN
    6.              ENTRY F(X)
    7.              IF (X .LT. 1.E-3) GO TO 200
    8.              U2 = X*X
    9.              V = POT(1./X)/E
   10.              WORK = B*B*U2 + V
   11.              IF (WORK .LT. 0.01) GO TO 100
   12.              IF (WORK .GT. 0.9999999.AND. WORK .LT. 1.01) WORK = 0.9999999
   13.              VEFF = (1. - SQRT(1. - WORK))/U2
   14.              RETURN
   15.      100     VEFF = 0.5*(B*B+V/U2)*(1. + 0.25*WORK*(1. + 0.5*WORK))
   16.              RETURN
   17.      200     VEFF = 0.5*B*B
   18.              RETURN
   19.              ENTRY F2(X)
   20.              U = B*X
   21.              WORK = 1. - U*U - POT(1./X)/E
   22.              IF (WORK .LT. 0. .AND. WORK .GT. -0.01) WORK = -WORK
   23.              VEFF = SQRT(WORK)/(X*X)
   24.              RETURN
   25.              ENTRY F3(X)
   26.              IF (X .LT. 1.E-3) GO TO 400
   27.              U2 = X*X
   28.              BB = B*B
   29.              DODA = BB + POT(1./X) /(U2*E)
   30.              WORK = U2*DODA
   31.              IF (WORK .LT. 0.01) GO TO 300
   32.              VEFF = (SQRT(ABS(1.-WORK))-SQRT(ABS(1.-BB*U2)))/U2
   33.              RETURN
   34.      300     WORK = DODA*DODA
   35.              V = BB*BB
   36.              VEFF = 0.5*(BB-DODA) + 0.125*U2*((V-WORK) + 0.5*U2*(BB*V-WORK*
   37.             1 DODA))
   38.              RETURN
   39.      400     VEFF = 0.
   40.              RETURN
   41.              END

END OF COMPILATION:          NO  DIAGNOSTICS.
```

C. POTIN/POT

This function is a user-defined routine which returns values of potential constants (POTIN) and values of V(r) (POT)

1. POTIN

CALL: SIGMA = POTIN (C6,EUNIT, RUNIT)

OUTPUT: C6 - values of C_6 in reduced units EUNIT = e and RUNIT = a

EUNIT - e dimensional unit in cpe of energy (of order ε)

RUNIT - a dimensional unit in $\mathring{A}$ of distance (of order r_m or σ)

POTIN - returns value of σ ($V(\sigma) = 0$) in units of RUNIT

2. POT

CALL: V = POT(R)

INPUT: R = r in reduced units

OUTPUT: POT - V(r) in reduced units

3. SAMPLE PROGRAM

The program listing is of a function which reads in a pointwise potential (EUNIT = ε and RUNIT = r_m). The value of V(r) is calculated by quadratic interpolation

$$V(r) \simeq V + Ar + Br^2 \qquad \text{(VII-9)}$$

where A(I) and B(I) are calculated by the routine. Extrapolation to low r is done by extrapolation of the closest quadratic, extrapolation to large r is done by

$$V(r) \simeq -\frac{C_6}{r^6} - \frac{C_8}{r^8} \qquad \text{(VII-10)}$$

where C8 = C_8 is calculated from the last data point and C_6.

INPUT: Card 1: NOB (I5)

Card 2: EPS, RM, C6, SIGMA (10F8.3)

Card 3ff: (X(I), V(I), I = 1,NOB) (10F8.3)

where

NOB - no. of points

EPS - ε (cpe)

RM - r_m ($\mathring{A}$)

C6 - C_6 (reduced units)

SIGMA - σ (reduced units)

X(I) - I-th value of r (reduced units)

V(I) - I-th value of V (reduced units)

REMARKS: (1) C_8 is calculated from $C_8 = - r^8V - r^6C_6$ for r = X(NOB), V = V(NOB)

(2) It is not necessary that EPS = ϵ and RM = r_m, but any scale factors may be used.

```
 POT
*05/31/72*01.53.25  (.0)      POT
    1.              FUNCTION POTIN(C6CONS,EPS,RM)
    2.              DIMENSION X(50),V(50),A(50),B(50)
    3.              READ (5,10) NOB
    4.        10    FORMAT (I5)
    5.              READ (5,20) EPS,RM,C6,SIGMA
    6.              C6CONS = C6
    7.        20    FORMAT (10F8.3)
    8.              READ (5,20) (X(I),V(I),I=1,NOB)
    9.              R2 = X(NOB)*X(NOB)
   10.              R6 = R2*R2*R2
   11.              C8 = -R2*(V(NOB)*R6 + C6)
   12.              R2 = 1./X(NOB-1)**2
   13.              R6 = R2*R2*R2
   14.              TEST = -R6*(R2*C8 + C6)/V(NOB-1)
   15.              NN = NOB - 1
   16.              DO 90 I = 2,NN
   17.              H1 = X(I) - X(I-1)
   18.              H2 = X(I+1)-X(I)
   19.              HH = X(I+1)-X(I-1)
   20.              D12 = V(I) - V(I-1)
   21.              D23 = V(I+1) - V(I)
   22.              H12 = H1/H2
   23.              A(I) = (H12*D23 + D12/H12)/HH
   24.        90    B(I) = (H12*D23 - D12)/(H1*HH)
   25.              NN = (NOB+1)/2
   26.              RUP = X(NOB)
   27.              POTIN = SIGMA
   28.              RETURN
   29.              ENTRY POT(R)
   30.              IF (R .GT. RUP) GO TO 500
   31.              I = NN
   32.       100    IF (R .GT. X(I)) GO TO 200
   33.              I = (I+1)/2
   34.              IF (I .GT. 1) GO TO 100
   35.              I = 2
   36.              GO TO 400
   37.       200    I = I + 1
   38.              IF (I .EQ. NOB) GO TO 300
   39.              IF (R-X(I)) 400,400,200
   40.       300    I = NOB - 1
   41.       400    HH = R - X(I)
   42.              POTIN = V(I) + HH*(A(I)+HH*B(I))
   43.              RETURN
   44.       500    R2 = 1./(R*R)
   45.              R6 = R2*R2*R2
   46.              POTIN = -R6*(R2*C8+C6)
   47.              RETURN
   48.              END

D OF COMPILATION:          NO  DIAGNOSTICS.
```

D. QJBVAL

Function which computes $\int_b^\infty b' \sin^2 \eta_\ell^{JB} db'$ where the integral is done by expansion of $\sin^2$ and $\eta_\ell^{JB} = \frac{3\pi}{32} \frac{C_6 A}{E b^5}$.

CALL: Q = QJBVAL (CJB,B)

INPUT: CJB - $3\pi AC_6/32E$

B - b

OUTPUT: QJBVAL - $\int_b^\infty b' \sin^2 \eta^{JB} db'$

REMARKS: The value returned is correct to more than 3 significant figures for values of B for which $\eta_\ell^{JB} \leq 2$.

```
QJBVAL
*05/31/72*01153127 (,0)      QJBVAL
   1*              FUNCTION QJBVAL(CJB,B)
   2*        C     PERFORMS INTEGRAL FROM B TO INFINITY OF B*(SIN ETAJB)**2
   3*        C     WHERE ETAJB IS J-B PHASE SHIFT FOR LONG-RANGE C6*
   4*        C     ETAJB = CJB/B**5, CJB = 3*PI*C6/(32*E)
   5*              DIMENSION S(10)
   6*              DATA S/1.,-0.33333333,0.044444444,-0.31746031E-2,0.14109347E-3
   7*             1 -.42755597E-5,.93968345E-7,-.15661391E-8,.20472406E-10,
   8*             2 -.21549902E-12/
   9*              QJBVAL = 0.
  10*              T = B*B
  11*              Z = CJB/(T*T)
  12*              Z = Z*Z
  13*              T = Z/T
  14*              DODA = 1.
  15*              DO 500 I = 1,10
  16*              DELQ = S(I)*DODA/(10*I-2)
  17*              QJBVAL = QJBVAL + DELQ
  18*              IF (ABS(DELQ) .LT. 1.E-3*QJBVAL) GO TO 600
  19*        500   DODA = DODA*T
  20*        600   QJBVAL = Z*QJBVAL
  21*              RETURN
  22*              END

D OF COMPILATION:        NO  DIAGNOSTICS.
```

E. DNTERP

This function performs finite difference interpolation based on a difference table constructed by DIFTAB (q.v.) The appropriate formula (Forward, Backward, or Stirling's central difference) is used with maximum order such that:

(1) The order is less than or equal to the size of the difference table.

(2) The size of the next term to be added is less than the propagated error.

CALL: VALUE = DNTERP (X, XO, H, N, D, ND, EPS)

ARGUMENTS: X - abscissa of interpolant

XO - left-most abscissa of data

H - stepsize of data

Y(I) - I-th ordinate of data

D - difference table from DIFTAB (q.v.)

ND - highest-order difference available in D (returned by DIFTAB, q.v.)

EPS - estimate of the absolute error of a Y(I)

OUTPUT: DNTERP - value of interpolated ordinate corresponding to X .

REMARK: The program extrapolates beyond the range of the data by means of the nearest forward/backwards difference polynomial. (This could be improved by extrapolating the difference table.)

```
1 DNTERP
)=05/31/72=01153; 18 (,0)      DNTERP
    1.              FUNCTION DNTERP(X,XO,H,Y,N,D,ND,EPS)
    2.        C     PERFORMS DIFFERENCE INTERPOLATION UP TO ORDER ND
    3.        C     D = DIFFERENCE TABLE FROM DIFTAB
    4.        C     EPS = ESTIMATE OF ABSOLUTE ERROR IN Y'S
    5.              DIMENSION Y(1),D(1)
    6.              M = 0
    7.              DODA = 1.
    8.              TOL = EPS
    9.              THETA = (X-XO)/H
   10.              I = THETA + 0.5
   11.              IF (I .LT. 0) I = 0
   12.              IF (I .GE. N) I = N - 1
   13.              THETA = THETA - I
   14.              I = I + 1
   15.              DNTERP = Y(I)
   16.              IF (I .GT. N-ND) GO TO 2000
   17.              IF (I .LT. ND) GO TO 1000
   18.        C     USE STIRLING'S CENTRAL DIFFERENCE FORMULA
   19.              DO 400 J = 1,ND
   20.              TOL = 2.*TOL
   21.              L = J/2
   22.              K = M + I - L
   23.              IF (ABS(D(K)) .LE. TOL) GO TO 500
   24.              DODA = DODA/J
   25.              IF (2*L .NE. J) GO TO 300
   26.              TERM = THETA*DODA
   27.              DODA = DODA*(THETA-L)
   28.              GO TO 350
   29.        300   DODA = DODA*(THETA+L)
   30.              DNTERP = DNTERP + 0.5*DODA*(D(K) + D(K-1))
   31.              GO TO 400
   32.        350   DNTERP = DNTERP + TERM*D(K)
   33.        400   M = M + N - J
   34.        500   RETURN
   35.        C     USE NEWTON'S FORWARD DIFFERENCE FORMULA
   36.        1000  DO 1500 J = 1,ND
   37.              TOL = 2.*TOL
   38.              K = M + I
   39.              IF (ABS(D(K)) .LE. TOL) GO TO 1600
   40.              DODA = DODA*(THETA-J+1)/J
   41.              DNTERP = DNTERP + DODA*D(K)
   42.        1500  M = M + N - J
   43.        1600  RETURN
   44.        C     USE NEWTON'S BACKWARDS DIFFERENCE FORMULA
   45.        2000  DO 2500 J = 1,ND
   46.              TOL = 2.*TOL
   47.              K = M + I - J
   48.              IF (ABS(D(K)) .LE. TOL) GO TO 2600
   49.              DODA = DODA*(THETA+J-1)/J
   50.              DNTERP = DNTERP + DODA*D(K)
   51.        2500  M = M + N - J
   52.        2600  RETURN
   53.              END

END OF COMPILATION:          NO  DIAGNOSTICS.
```

F. DIFTAB

This program constructs a finite difference table from N equally-spaced data points, with ordinates y_i $(1 \leq i \leq N)$. Successively higher order differences are computed until no element of the current column in the difference table is larger than the propagated error.

CALL DIFTAB (Y, N, D, ND, EPS)

INPUT: Y(I) - y_i I-th ordinate

N - no. data points

EPS - estimate of absolute errors in Y(I)

OUTPUT: D - difference table stored in packed-mode: i.e., the first-order differences are in D(1) ... D(N-1), the second-order differences in D(N) ... D(2N-2), etc.

ND - set to highest-order difference calculated.

```
DIFTAB
=05/31/72=0115312O (,0)      DIFTAB
    1.              SUBROUTINE DIFTAB(Y,N,D,ND,EPS)
    2.        C     CALCULATES DIFFERENCE TABLE OF Y(1) ,.. Y(N)
    3.        C     NOTE = D MUST BE DIMENSIONED AT LEAST ND*(N-1)
    4.        C     ND = UPON ENTRY, MAX ORDER DIFFERENCE TO BE CALCULATED. UPON
    5.        C     RETURN IS ACTUAL HIGHEST ORDER
    6.        C     EPS = ABSOLUTE ERROR ESTIMATE OF Y'S TO STOP DIFFERENCES
    7.              DIMENSION Y(1),D(1)
    8.              MAXD = ND
    9.              ERRLVL = 2.*EPS
   10.              M = N
   11.              T = FORDIF (Y,M,D)
   12.              ND = 1
   13.              K = 1
   14.        100   IF (ND .GE. MAXD .OR. M .LE. 1 .OR. T .LE. ERRLVL) GO TO 200
   15.              ND = ND + 1
   16.              ERRLVL = 2.*ERRLVL
   17.              J = K + M
   18.              T = FORDIF(D(K),M,D(J))
   19.              K = J
   20.              GO TO 100
   21.        200   RETURN
   22.              END

ND OF COMPILATION:          NO  DIAGNOSTICS.
```

G. FORDIF

Function which calculates the vector of forward-differences corresponding to a vector of ordinates.

CALL: DMAX = FORDIF (Y, N, D)

INPUT: Y(I) - I-th ordinate

N - No. of data

OUTPUT: D(I) - I-th forward difference, Y(I+1) - Y(I)

FORDIF - $\max|D(I)|$, $1 \leq I \leq N-1$

```
 FORDIF
=05/31/72=01:53:21 (,0)      FORDIF
    1.              FUNCTION FORDIF (Y,N,D)
    2.        C     CALCULATES FORWARD DIFFERENCES D(I) = Y(I+1) - Y(I)
    3.        C     RETURNS VALUE OF MAXIMUM DIFFERENCE
    4.              DIMENSION Y(1),D(1)
    5.              N = N - 1
    6.              T = 0.
    7.              DO 100 I = 1,N
    8.              D(I) = Y(I+1) - Y(I)
    9.        100   IF (T .LT. ABS(D(I)) ) T = ABS(D(I))
   10.              FORDIF = T
   11.              RETURN
   12.              END

NO OF COMPILATION:          NO DIAGNOSTICS.
```

H. EPSALG

Subroutine which extrapolates a sequence of partial sums to a limit by means of the ε-algorithm*. Only the diagonal of the matrix of con-

- - - - - - - - - - -

* See P. Wynn, "Five Lectures on the Numerical Application of Continued Fractions", Orientation Lecture Series No. 5, Mathematics Research Center, University of Wisconsin, Madison, Wisconsin, 53706.

- - - - - - - - - - -

vergents is stored.

The algorithm is as follows to find the limit of a sequence of partial sums S_m (m = 0, 1, 2, ...):

$$\varepsilon_{-1}^{(m)} = 0 \qquad (m = 1, 2, \ldots)$$

$$\varepsilon_{0}^{(m)} = S_m \qquad (m = 0, 1, 2, \ldots)$$

$$\varepsilon_{r+1}^{(m)} = \varepsilon_{r-1}^{(m+1)} + 1/(\varepsilon_{r}^{(m+1)} - \varepsilon_{r}^{(m)}) \qquad (r = 1, 2, \ldots)$$

The $\varepsilon_{2r}^{(m)}$ are improved estimates of the limit.

CALL EPSALG (S, M, X)

INPUT: S - current value of partial sum

M - Number of times EPSALG has been called previously for this series

OUTPUT: M - will be replaced by M + 1

X(1) ... X(M) - will contain the ε-diagonal of convergents.

REMARKS: The approximations to the limit of the series will be in every other X(I) . The best value for the limit will be found in X(1) if M is odd or X(2) if M is even.

```
 EPSALG
=05/31/72=01153122 (,0)      EPSALG
    1.              SUBROUTINE EPSALG(S,M,X)
    2.        C     EXTRAPOLATES SUM OF SERIES S BY EPSILON ALGORITHM
    3.        C     S IS VALUE OF CURRENT PARTIAL SUM, AT EXIT M IS LENGTH OF
    4.        C     THE EPSILON DIAGONAL X(1) ...X(M)
    5.        C     BEST ESTIMATE OF SUM IS IN X(J), WHERE J = 2*MOD(M,2)
    6.              DIMENSION X(1)
    7.              K = M
    8.              M = M + 1
    9.              X(M) = 0.
   10.              A1 = S
   11.        100   IF (K .LE. 0) GO TO 200
   12.              A0 = X(K+1) + 1./(A1-X(K))
   13.              X(K+1) = A1
   14.              A1 = A0
   15.              K = K - 1
   16.              GO TO 100
   17.        200   X(1) = A1
   18.              RETURN
   19.              END

ND OF COMPILATION:          NO  DIAGNOSTICS.
```

I. REGFAL

This subroutine finds a root of the equation $F(x) = 0$, given an interval which contains the root, by a strategy using regula falsi and the method of bisection. Barring multiple roots or excessive rounding-error, the routine guarantees convergence. The algorithm proceeds as follows:

(1) Bisect the original interval until a sign change is found. (If none is found, error return.)

(2) Apply regula falsi (secant method) until convergence is attained.

(a) If the change in interval length at any step is less than 10% of the previous interval length, apply the method of bisection once.

(b) If convergence is not attained, or the root is lost, error return.

CALL REGFAL (XL, XU, X, F, EPS, LIM1, LIM2, IER)

INPUT: XL,XU - Endpoints of an interval containing the root

F - name of function which stores the value of $F(x)$ in y in the call Y = F(X)

EPS - relative error criterion for root. Termination occurs when relative change in root is less than EPS or the root itself is less than EPS

LIM1 - Limit on number of applications of bisection to find a sign change; if LIM1 = 0, Step (1) above is skipped

LIM2 - limit on number of applications of regula falsi in step (2)

OUTPUT: X - estimate of root

IER - 0 Normal

1 No root or a multiple root in interval (to tolerance EPS)

2 No sign change found with LIM1 iterations

3 No convergence to EPS tolerance with LIM2 iterations

4 Root was lost in regula falsi iterations (no longer a sign change)

REMARK: The convergence testing could be improved slightly by adding

line 12.5 HO = EPS*H

and replacing line 50 with

```
IF (ABS(DELX) . LT . HO) GO TO 1000
```

```
REGFAL
-05/31/72-01:53:24 (,0)      REGFAL
    1.              SUBROUTINE REGFAL (XL,XU,X,F,EPS,LIM1,LIM2,IER)
    2.        C     FINDS ROOT OF THE EQTN F(X) = 0 IN INTERVAL (XL,XU)
    3.        C     BY METHOD OF REGULA FALSI (GUARANTEED CONVERGENCE)
    4.        C     EPS = RELATIVE ERROR TOLERANCE FOR ROOT
    5.        C     LIM1 = ALLOWABLE ITER. TO FIND SIGN CHANGE IN AN INTERVAL
    6.        C     LIM2 = ALLOWABLE ITERATIONS TO FIND ROOT
    7.        C     IER = 0,NORMAL.  = 1 NO OR MULT ROOT TO TOLERANCE EPS.  = 2 NO
    8.        C     SIGN CHANGE WITH LIM1 ITERATIONS.  = 3 NO CONVERGENCE WITH LIM2 IT
    9.        C       = 4, LOST ROOT IN REG. FALSI ITERATIONS
   10.              FA = F(XL)
   11.              X = XL
   12.              H = XU - XL
   13.              IF (LIM1 .EQ. 0) GO TO 300
   14.              J = 1
   15.              HH = EPS*H
   16.              DO 200 I = 1,LIM1
   17.              IF (H .LE. HH) GO TO 250
   18.              X = XL
   19.              DO 100 K = 1,J
   20.              FB = F(X+H)
   21.              IF (FA*FB .LE. 0.) GO TO 350
   22.       100    X = X + H
   23.              H = 0.5*H
   24.       200    J = 2*J
   25.              IER = 2
   26.              RETURN
   27.       250    IER = 1
   28.              RETURN
   29.       300    FB = F(X+H)
   30.       350    A = X
   31.              B = X + H
   32.              FA = F(A)
   33.              HH = ABS(H)
   34.              XP = A
   35.              DO 800 I = 1,LIM2
   36.              X = A - FA*H/(FB-FA)
   37.              J = 1
   38.       400    FX = F(X)
   39.              IF (FX*FA ) 500,1000,600
   40.       500    FB = FX
   41.              B = X
   42.              GO TO 700
   43.       600    FA = FX
   44.              A = X
   45.              IF (FA*FB .GT. 0.) GO TO 900
   46.       700    H = B - A
   47.              IF (J .EQ. 2) GO TO 800
   48.              DELX = X - XP
   49.              XP = X
   50.              IF (ABS(DELX) .LT. EPS*ABS(X) .OR. ABS(X) .LT. EPS) GO TO 1000
   51.              IF (ABS(H) .LT. 0.9*HH) GO TO 800
   52.              X = 0.5*(A+B)
   53.              J = 2
   54.              GO TO 400
   55.       800    HH = ABS(H)
```

```
   56.              IER = 3
   57.              RETURN
   58.        900   IER = 4
   59.              RETURN
   60.       1000   IER = 0
   61.              RETURN
   62.              END
```

```
ND OF COMPILATION:      NO  DIAGNOSTICS.
```

J. ADSIMP

This function returns the value of the integral

$$\int_{XL}^{XU} F(X)\,dX$$

obtained by means of adaptive application of Simpson's rule. The program is a slightly modified version of Alg. 182 , Comm. ACM 6, 315 (1963).

CALL: VALUE = ADSIMP (XL, XU, F, ERRTOL, MAX, IER)

INPUT: XL, XU - Endpoints of the interval of integration

F - name of function which stores the value of the integrand in Y from the call Y = F(X)

ERRTOL - Error bound on absolute error of the integral

MAX - maximum number of integration points which may be used

OUTPUT: IER - if 0, normal convergence. If $\neq$ 0, then more than MAX points are needed for convergence.

ADSIMP - Returned value of integral

REMARK: All of the dimensions in lines 6-8 could be safely reduced to 15. (which necessitates a 15 in line 41)

```
ADSIMP
05/31/72-01:53:28 (,0)      ADSIMP
   1.            FUNCTION ADSIMP(XL,XU,F,ERRTOL,MAX,IER)
   2.      C     INTEGRATES F(X) FROM XL TO XU BY ADAPTIVE SIMPSONS RULE
   3.      C     ERRTOL = ALLOWABLE ABSOLUTE ERROR TOLERANCE
   4.      C     MAX = LIMIT ON NO FUNCTION CALLS
   5.      C     IF IER .NE. 0 MORE THAN MAX POINTS ARE REQUIRED
   6.            DIMENSION DX(30),EPSP(30),X2(30),X3(30),F2(30),F3(30),F4(30)
   7.           1FMP(30),FBP(30),EST2(30),EST3(30),PVAL(30,3)
   8.            INTEGER RTRN(30)
   9.            EPS = ERRTOL
  10.            A = XL
  11.            B = XU
  12.            LVL = 0
  13.            EST = 1.0
  14.            DA = B - A
  15.            FA = F(A)
  16.            FM = 4.0*F((A+B)*0.5)
  17.            FB = F(B)
  18.            KOUNT = 3
  19.      100   LVL = LVL + 1
  20.            DX(LVL) = 0.333333333*DA
  21.            SX = DX(LVL)*0.166666667
  22.            F1 = 4.0*F(A + 0.5*DX(LVL))
  23.            X2(LVL) = A + DX(LVL)
  24.            F2 (LVL) = F(X2(LVL))
  25.            X3(LVL) = X2(LVL) + DX(LVL)
  26.            F3(LVL) = F(X3(LVL))
  27.            EPSP(LVL) = EPS
  28.            F4(LVL) = 4.0*F(X3(LVL) + 0.5*DX(LVL ))
  29.            FMP(LVL) = FM
  30.            EST1 = SX*(FA + F1 + F2(LVL))
  31.            FBP(LVL) = FB
  32.            EST2(LVL) = SX*(F2(LVL) + F3(LVL) + FM)
  33.            EST3(LVL) = SX*(F3(LVL) + F4(LVL) + FB)
  34.            SUM = EST1 + EST2(LVL) + EST3(LVL)
  35.            KOUNT = KOUNT + 4
  36.            IF (ABS(EST-SUM) .GT. EPSP(LVL) ) GO TO 500
  37.      400   LVL = LVL - 1
  38.            I = RTRN (LVL)
  39.            PVAL (LVL,I) = SUM
  40.            GO TO (600,700,800),I
  41.      500   IF (LVL .GE. 30 .OR. KOUNT .GE. MAX) GO TO 400
  42.            RTRN (LVL) = 1
  43.            FM = F1
  44.            FB = F2(LVL)
  45.            EST = EST1
  46.      550   DA = DX(LVL)
  47.            EPS = 0.577*EPSP(LVL)
  48.            GO TO 100
  49.      600   RTRN(LVL) = 2
  50.            FA = F2(LVL)
  51.            FM = FMP(LVL)
  52.            FB = F3(LVL)
  53.            EST = EST2(LVL)
  54.            A = X2(LVL)
  55.            GO TO 550
```

```
 56.      700    RTRN (LVL) = 3
 57.             FA = F3(LVL)
 58.             FM = F4(LVL)
 59.             FB = FBP(LVL)
 60.             EST = EST3(LVL)
 61.             A = X3(LVL)
 62.             GO TO 550
 63.      800    SUM = PVAL(LVL,1) + PVAL(LVL,2) + PVAL(LVL,3)
 64.             IF (LVL .GT. 1) GO TO 400
 65.             ADSIMP = SUM
 66.             IER = 0
 67.             IF (KOUNT .GT. MAX) IER = 1
 68.             RETURN
 69.             END

ID OF COMPILATION:          NO  DIAGNOSTICS.
```

K. SAMPLE OUTPUT

The problem shown is for the Na-Hg system with the pointwise potential taken from U. Buck and H. Pauly, J. Chem. Phys. 54, 1929 (1971). The units of energy and length were $\varepsilon = 8.79$ cpe and $r_m = 4.72$ Å, resp. The output shown is for $E^* = 1.65$.

NOTES: (1) The last partial wave calculated was at $\ell = 455$, ($b^* = 3$), when $\delta Q < 0.1$ Å^2/b.

(2) The convergents of the ε-algorithm after 5 extrapolations were:

= 175.4996 (best), 175.4999, 175.0902.

(3) The calculation took 6 seconds on an Univac 1108 (1.5 μs add time). Since every seventh η_ℓ was calculated, without interpolation the execution would have taken 40 seconds.

(4) The ε-algorithm extrapolated cross section was 696.6 Å^2, with $C_6^{app} = 0.6690 \times 10^5$ cpe - Å^6.

ENTRY TO JWKB/DNTERP

TOLERANCE FOR YO (RELATIVE) .10000-04
TOLERANCE FOR ETA (ABS) .50000-02
TOLERANCE FOR JB LIMIT .10000-01

BUCK-PAULY N_A-HG 45-POINT INTERPOLATED POTENTIAL

PHASE SHIFTS AND CROSS SECTIONS FOR REDUCED MASS .342513+02
LONG-RANGE C6 .500000+00
ZERO OF POTENTIAL .810840+00
UNITS OF ENERGY AND LENGTH .879000+01 .472000+01

ENERGY 1.65000 WAVE NO. .298886+02 A=WAVE*RUNIT .141074+03

FOLLOWING TABLE IN REDUCED UNITS EUNIT,RUNIT

L	B	YO	RO	ETA	ETAJB	DELQ	IER	ISW
0	.003544	1.422920	.702780	-.856023+02	.225136+14	.175033-02	0	0
7	.053163	1.421865	.703302	-.749108+02	.296475+08	.116608-01	0	0
14	.102783	1.418988	.704728	-.647499+02	.109763+07	.908811-01	0	0
21	.152402	1.414354	.707037	-.551184+02	.153145+06	.149412+00	0	0
28	.202021	1.408074	.710190	-.460106+02	.374171+05	.162609+00	0	0
35	.251640	1.400376	.714094	-.374170+02	.124782+05	.195012-01	0	0
42	.301260	1.391112	.718849	-.293494+02	.507399+04	.233086+00	0	0
49	.350879	1.380208	.724528	-.217882+02	.236738+04	.142485-01	0	0
56	.400498	1.368192	.730892	-.147260+02	.122195+04	.276974+00	0	0
63	.450118	1.355286	.737852	-.816297+01	.681437+03	.408493+00	0	0
70	.499737	1.341338	.745524	-.209508+01	.403969+03	.374505+00	0	0
77	.549356	1.326383	.753930	.347989+01	.251643+03	.605077-01	0	0
84	.598975	1.310477	.763081	.855836+01	.163309+03	.347806+00	0	0
91	.648595	1.293687	.772984	.131465+02	.109695+03	.194877+00	0	0
98	.698214	1.276065	.783659	.172409+02	.758775+02	.697211+00	0	0
105	.747833	1.257647	.795136	.208403+02	.538309+02	.623540+00	0	0
112	.797452	1.238331	.807538	.239393+02	.390420+02	.689164+00	0	0
119	.847072	1.218499	.820682	.265423+02	.288705+02	.825251+00	0	0
126	.896691	1.198434	.834422	.286279+02	.217191+02	.107484+00	0	0
133	.946310	1.176721	.849819	.301672+02	.165916+02	.851474+00	0	0
140	.995929	1.154330	.866303	.311727+02	.128503+02	.577642-01	0	0
147	1.045549	1.130917	.884238	.316020+02	.100771+02	.357779-01	0	0
154	1.095168	1.105848	.904284	.314321+02	.799193+01	.286874-03	0	0
161	1.144787	1.079500	.926355	.306323+02	.640371+01	.570406+00	0	0
168	1.194406	1.050014	.952368	.291102+02	.517958+01	.657322+00	0	0
175	1.244026	1.017124	.983165	.268476+02	.422580+01	.121839+01	0	0
182	1.293645	.978690	1.021774	.237233+02	.347520+01	.126025+01	0	0
189	1.343264	.929633	1.075693	.197253+02	.287904+01	.792292+00	0	0
196	1.392883	.864247	1.157077	.149959+02	.240149+01	.594686+00	0	0
203	1.442503	.790788	1.264561	.103836+02	.201591+01	.966429+00	0	0
210	1.492122	.725005	1.379301	.707472+01	.170229+01	.755222+00	0	0
217	1.541741	.684871	1.460130	.527661+01	.144543+01	.110085+01	0	0
224	1.591360	.654228	1.528519	.405952+01	.123371+01	.100412+01	0	0
231	1.640980	.628806	1.590316	.313450+01	.105813+01	.826042-04	0	0
238	1.690599	.604714	1.653675	.237284+01	.911701+00	.817165+00	0	0
245	1.740218	.584066	1.712136	.185593+01	.788927+00	.160253+01	0	0
252	1.789837	.565683	1.767773	.147790+01	.685468+00	.177444+01	0	0
259	1.839457	.548497	1.823165	.117491+01	.597871+00	.156592+01	0	0

266	1.889076	.533001	1.876168	.991833+00	.523369+00	.132352+01	0	0
273	1.938695	.518827	1.927424	.855278+00	.459735+00	.110438+01	0	0
280	1.988314	.505414	1.978574	.734174+00	.405163+00	.892486+00	0	0
287	2.037934	.492703	2.029621	.632512+00	.358183+00	.712228+00	0	0
294	2.087553	.480683	2.080371	.548260+00	.317591+00	.567089+00	0	0
301	2.137172	.469277	2.130939	.477267+00	.282395+00	.450956+00	0	0
308	2.186791	.458430	2.181358	.417145+00	.251778+00	.358958+00	0	0
315	2.236411	.448098	2.231656	.365977+00	.225059+00	.286406+00	0	0
322	2.286030	.438240	2.281854	.322221+00	.201672+00	.229249+00	0	0
329	2.335649	.428822	2.331968	.284637+00	.181141+00	.184175+00	0	0
336	2.385268	.419813	2.382014	.252239+00	.163068+00	.148570+00	0	0
343	2.434888	.411184	2.432001	.224201+00	.147116+00	.120355+00	0	0
350	2.484507	.402911	2.481939	.199832+00	.133001+00	.978999-01	0	0
357	2.534126	.394970	2.531835	.178604+00	.120480+00	.799813-01	0	0
364	2.583745	.387342	2.581696	.160043+00	.109347+00	.656165-01	0	0
371	2.633365	.380008	2.631527	.143761+00	.994260-01	.540507-01	0	0
378	2.682984	.372949	2.681332	.129441+00	.905659-01	.447026-01	0	0
385	2.732603	.366151	2.731115	.116801+00	.826366-01	.371101-01	0	0
392	2.782222	.359599	2.780879	.105632+00	.755259-01	.309289-01	0	0
399	2.831842	.353279	2.830626	.957176-01	.691369-01	.258658-01	0	0
406	2.881461	.347179	2.880358	.869143-01	.633857-01	.217121-01	0	0
413	2.931080	.341288	2.930078	.790664-01	.581991-01	.182855-01	0	0
420	2.980699	.335594	2.979788	.720631-01	.535136-01	.154523-01	0	0
427	3.030319	.330089	3.029487	.657892-01	.492735-01	.130970-01	0	0
434	3.079938	.324762	3.079179	.601676-01	.454303-01	.111364-01	0	0
441	3.129557	.319605	3.128863	.551144-01	.419412-01	.949672-02	0	0
448	3.179176	.314610	3.178540	.505678-01	.387688-01	.812256-02	0	0
455	3.228796	.309769	3.228217	.464612-01	.358800-01	.696481-02	0	0

MAINING CROSS SECTION IS LESS THAN 0.05 SQ ANG

SILON ALGORITHM CONVERGES ...

175.4996 -54.9791 175.4999 102.8419 175.0902

ME (SEC) 5.937

CROSS SECTION AT LAST PHASE SHIFT 694.92 Q 175.090

JB EXTRAPOLATED CROSS SECTION (SQ ANG) 696.51 QJB 175.490 C6 APPARENT .668939

HIGHEST ORDER DIFFERENCE USED IN INTERPOLATION 10

EPSILON ALGORITHM CROSS SECTION 696.55 Q 175.500 C6 APPARENT .669031+05