AAEC/E547

AAEC/E547

AUSTRALIAN ATOMIC ENERGY COMMISSION

RESEARCH ESTABLISHMENT

LUCAS HEIGHTS RESEARCH LABORATORIES

MASTER - A COMPUTER PROGRAM SOLUTION OF THE MASTER EQUATIONS FOR LASER MULTI-PHOTON DISSOCIATION

by

R.B. KNOTT

September 1982

ISBN 0 642 59751 0

AUSTRALIAN ATOMIC ENERGY COMMISSION
RESEARCH ESTABLISHMENT

LUCAS HEIGHTS RESEARCH LABORATORIES

MASTER - A COMPUTER PROGRAM SOLUTION OF THE MASTER EQUATIONS FOR LASER MULTI-PHOTON DISSOCIATION

by

R.B. KNOTT

ABSTRACT

The program MASTER was written to analyse the experimental data on the laser multi-photon dissociation of ethyl acetate, and of uranyl and vanadyl diketonates. The program was written in FORTRAN IV and executed on an IBM3031 computer. A typical calculation of fraction of molecules dissociated requires approximately 2 megabytes of storage and an execution time of approximately 1-2 minutes per time step for a grain size of one quarter the laser frequency. Appendices are presented which contain details of the program coding.

National Library of Australia card number and ISBN 0 642 59751 0

The following descriptors have been selected from the INIS Thesaurus to describe the subject content of this report for information retrieval purposes. For further details please refer to IAEA-INIS-12 (INIS: Manual for Indexing) and IAEA-INIS-13 (INIS: Thesaurus) published in Vienna by the International Atomic Energy Agency.

ACETATES; ALGORITHMS; DIFFERENTIAL EQUATIONS; DISSOCIATION; FORTRAN; INTEGRAL EQUATIONS; KETONES; LASER RADIATION; M CODES; MOLECULES; URANYL COMPOUNDS; VANADIUM COMPOUNDS

CONTENTS

1. INTRODUCTION

The standard master equation describes the laser multi-photon dissociation of reactant molecules incorporating reactant/buffer collisions during the pulse. This technique was applied to the analysis of experimental data obtained on ethyl acetate, and uranyl and vanadyl diketonates. The assumptions of homogeneous absorption of the laser energy, and internal equilibrium of the reactant molecules proved reasonable.

The master equation technique uses the Gear [1971] algorithm to solve many coupled partial integro-differential equations. We have restricted the program to one channel dissociation since this was adequate to analyse the experimental data. Parameters required for this calculation are the energy dependent microscopic reaction rate coefficients, $k(E)$, for the reaction channel, and the density of states, $\rho(E)$, for the reactant molecule. Conventional RRKM theory [Robinson and Holbrook 1972] without angular momentum effects was used for these parameters. Also, we needed the probability per collision of reactant/buffer energy transfer from energy E' to energy E,P(E,E'), and the various experimental parameters.

A brief description of the theory will illustrate the assumptions made and indicate possible areas of change for different experimental conditions.

2. THEORY

If the reactant population function is $g(E,t)$, then

$$\frac{\partial g(E,t)}{\partial t} = \omega \int [P(E,E')g(E') - P(E',E)g(E)]\, dE' - k(E)g(E)$$
$$- [L^A(E) + L^S(E)]g(E) + L^A(E-h\nu)g(E-h\nu)$$
$$+ L^S(E+h\nu)g(E+h\nu) \qquad (1)$$

where

$P(E,E')$ ≡ probability of collisional transition from E' to E

ω $\equiv$ gas kinetic collision rate = $p_2 * 4.14\ E7 * d_c^2 / \sqrt{T_o \frac{m_1 m_2}{m_1+m_2}}$

p_2 $\equiv$ pressure of buffer

d_c $\equiv$ collision diameter

T_o $\equiv$ initial temperature

m_1, m_2 $\equiv$ mass of reactant, mass of buffer

$k(E)$ $\equiv$ (RRKM) microscopic rate coefficients

$L^A(E)$ $\equiv$ rate of absorption of radiation

$L^S(E)$ $\equiv$ rate of stimulated emission of radiation

$h\nu$ $\equiv$ laser frequency.

We need to solve equation (1) using finite difference techniques with grain size ΔE such that $h\nu = \ell\ \delta E$ where ℓ is an integer.

If, for more convenience, we adopt notation $g_i = g(E_i,t)$ for a given time, t, then equation (1) becomes

$$\frac{dg_i}{dt} = \omega \sum_j (P_{ij}\ g_j - P_{ji}\ g_i)\ \delta E - (k_i + L_i^A + L_i^S) g_i + L_{i-\ell}^A\ g_{i-\ell} + L_{i+\ell}^S\ g_{i+\ell} \quad . \tag{2}$$

The general form of P(E,E') is

$$P(E,E') = \text{func}(E-E') \quad \text{where } E<E'$$

and upward transitions are given by microscopic reversibility

$$P_{ij}\ \rho_i\ \exp(-E_j/k_BT) = P_{ji}\ \rho_i\ \exp(-E_i/k_BT) \tag{3}$$

where

k_B = Boltzmann's constant

T = temperature at time, t

also

$$L_i^S \rho_i = L_{i-\ell}^A \rho_{i-\ell} \quad . \tag{4}$$

The transition probability term in equation (2) can be rewritten to refer only to downward transitions by use of equation (3) since

$$\sum_j (P_{ij} g_j - P_{ji} g_i) = \sum_{j<i} P_{ji} b_i \left(\frac{g_j}{b_j} - \frac{g_i}{b_i}\right) + \sum_{j>i} P_{ij} b_j \left(\frac{g_j}{b_j} - \frac{g_i}{b_i}\right)$$

where

$$b_i = \rho_i \exp(-E_i/k_B T).$$

The rate of absorption of radiation L_i^A is given by

$$L_i^A = I\sigma_i/h\nu$$

where

$I \equiv$ laser intensity such that laser fluence, $F = \int I dt$
$\sigma_i \equiv$ reactant absorption cross section $= \sigma_o \exp(-x\, n^\beta)$
$x, \beta \equiv$ input parameters
$n \equiv$ number of photons absorbed.

The set of differential equations given by equation (2) is coupled to the equation for the buffer temperature

$$\frac{dT}{dt} = \frac{a}{C_v} \sum_i \left[(E_i - \Delta H) k_i \hat{g}_i + 2 \sum_{j<i} \hat{b}_i P_{ji} (E_j - E_i) \left(\frac{\hat{g}_j}{\hat{b}_j} - \frac{\hat{g}_i}{\hat{b}_i}\right) \omega \delta E\right] \tag{5}$$

where $\Delta H \equiv$ enthalpy change for the reaction
$C_v \equiv$ constant volume specific heat
$\hat{g}_i \equiv \phi g_i / \Sigma_i g_i$ and $\hat{b}_i = \phi b_i / \Sigma_i b_i$
$\phi \equiv$ initial mole fraction of reactant in the mixture.

Note that C_v will vary with temperature and with fraction dissociated but this variation can be ignored for sufficiently dilute mixtures.

Thus equations (2) and (5) describe the absorption of laser radiation from time t = 0 until the end of the laser pulse. After the laser pulse, the reactant/buffer mixture is allowed to approach internal thermal equilibrium.

The fraction of reactant molecules dissociated up to time t is simply given by

$$f(t) = 1 - \frac{\sum_i g_i(t)}{\sum_i g_i(o)}$$

where $g_i(o)$ is the initial population distribution, which was a Boltzmann distribution in all our calculations. The calculation is terminated when the difference in fraction dissociated between time steps t_m and t_{m-1} is within the limit set by the input data, or when the total time is 10 μs, whichever occurs first.

3. PROGRAM EXECUTION

All programs are written in FORTRAN IV and the MASTER program was designed to execute on the AAEC IBM3031 computer.

The program RRKM [Gilbert 1980] creates a data file containing k(E), ρ(E) and other data concerning the basic dissociation reaction. The input data are described card by card in Appendix A, together with a sample input file and a description of the output data file. This program is executed only once for each reactant/buffer/grain size calculation. The program MASTER is then executed with the RRKM file as input, plus the details of the particular experiment. A sample input is given in Appendix A. Because of the full-format input routine SCAN, the input data are self-explanatory. A listing of MASTER is given in Appendix B. A brief description of the subroutine GEAR (NAGFLIB : 793/504 : Mk 5 : Nov 74) is given in Appendix C together with details of the input parameters set up by the main program.

4. CONCLUSIONS

The program MASTER was used with outstanding success to analyse experimental data on the multi-photon dissociation of ethyl acetate [Eberhardt et al. 1981a], uranyl diketonates [Eberhardt et al. 1981b] and vanadyl

diketonates [Eberhardt et al. 1981c]. The limitations of this method are illustrated in the paper by Eberhardt et al. [1981d] who used it to analyse data on vanadyl chloride.

5. ACKNOWLEDGEMENT

This program was written in collaboration with Dr R.G. Gilbert of the University of Sydney. Dr Gilbert supplied the program RRKM and the subroutines ENEXPT and GEAR.

6. REFERENCES

Eberhardt, J.E., Knott, R.B., Pryor, A.W. and Gilbert, R.G. [1981a] - Chem. Phys. (in press)

Eberhardt, J.E., Hoare, I.C., Johnson, D.A., Knott, R.B., Pryor, A.W. and Waugh, A.B. [1981b) - Chem. Phys. (in press)

Eberhardt, J.E., Johnson, D.A., Knott, R.B., Pryor, A.W. and Waugh, A.B. [1981c] - Chem. Phys. Letters (submitted).

Eberhardt, J.E., Johnson, D.A., Knott, R.B., Pryor, A.W. and Waugh, A.B. [1981d] - Chem. Phys. Letters (submitted)

Gear. C.W. [1971] - Numerical Initial Value Problems in Ordinary Differential Equations. Prentice Hall, N.Y.

Gilbert, R.G. [1980] - Private communication.

Robinson,P.J. and Holbrook, K.A. [1972] - Unimolecular Reactions. Wiley-Interscience, London.

APPENDIX A
PROGRAM RRKM

Card	Description
1	Title (20A4)
2	Number of points in numerical integration (I4) Laser frequency (F6.0) Number of times laser frequency subdivided (F2.0) Number of pressures (I2) Number of temperatures (I2) Number of channels (up to 5 but restricted to 1 for MASTER) (I2)
3	Number of frequencies for molecule (free) Number of frequencies for complex 1 (free)
4	Critical energy E_o (channel 1) (kcal mol^{-1}) (free)
5	Symmetry number of the molecule (free)
6	Symmetry number of the complex 1 (free)
7	Collision diameter (nm) (F6.1) Mass of reactant (amu) (F6.1) Mass of buffer (amu) (F6.1)
8	Frequencies and degeneracies of complex 1 (listed as: f1,d1,f2,d2,...) (I5,I2,...)
9	Frequencies and degeneracies of the molecule (I5,I2,...)
10	Pressures (torr) (free)
11	Temperature (k) (free)
12	Input option: $\leq$0 INC is calculated using the laser frequency and dividing factor >0 INC equals 100 ≠0 output file for master equation program is produced =0 no master equation file is produced (free)

NOTES: All frequencies in cm^{-1}.
INC is the energy increment used in the numerical integration.

(Continued)

```
SAMPLE INPUT FILE FOR RRKM PROGRAM
**********************************

::::::: ETHYL-ACETATE + NITROGEN BUFFER ::::::::::::::::::::::::::::::::
  80 1045.4. 1 1 1
 13,11
 46.452
 .333
 1.
   5.5  88.1    28.0
3000 7 2200 1 1450 5 1400 1 1310 2 1140 5  850 2  710 3  575 1
 400 5  353 3
3000 8 1740 1 1450 6 1390 1 1280 1 1150 5  900 3  780 3  640 1
 400 3  147 2  100 1   60 1
0.03
1750.
-1
```

RRKM output file for MASTER

1. Title
2. Grain size, laser frequency (n x grain size), (actual) laser frequency (cm^{-1})
3. Critical energy E_o (kcal mol^{-1})
4. Collision diameter (nm), mass of reactant (amu), mass of buffer (amu)
5. Number of k_i parameters, ℓ
6. Number of channels (1)
7. $k_i, \ldots k_\ell$
8. Number of ρ_i parameters, r
9. $\rho_i, \ldots \rho_r$

(Continued)

```
SAMPLE INPUT FILE FOR MASTER PROGRAM
************************************


NUMBER OF GI PLOTS 20
TIME FOR GI PLOT .5 1. 1.5 2. 2.5 3. 3.5 4. 4.5 5. 5.5 6. 6.5
7. 7.5 8. 8.5 9. 9.5 10. MICROSECONDS
CONVERGENCE ON FRACTION DISSOCIATED 0.00001
WRITE GI EVERY 10 ITERATION
GI ERROR FACTOR .01
GI MAXIMUM ERROR 1.E-4
DELH 12.2 KCAL/MOL
ALPHA 700.C WAVENUMBERS
CV 4.947 CAL/MOL/K
PRES .015 TORR REACTANT 3. TORR BUFFER
SIGO 3.75E-19 SQCM
XX .01
BETA 1.
DELTA TIME IN NUMERICAL INTEGRATION 10.E-9 S
LASER FLUENCE 3.7 J/SQCM
NUMBER OF STEPS IN LASER PULSE TIME PROFILE 20
LASER POWER FOR EACH STEP .06 .14 .17 .14 .08 .05 .045 .04
 .035 2*.03 2*.025 5*.02 2*.015
WIDTH FOR EACH STEP 20*100.E-9 S
```

APPENDIX B

SOLUTION OF MASTER EQUATION

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C                                                                              C
C  SOLUTION OF THE MASTER EQUATION                                             C
C  INCLUDES MOLECULAR COLLISIONS DURING THE LASER PULSE                        C
C  PROGRAM   WRITTEN IN COLLABORATION WITH DR R.GILBERT,                       C
C  DEPARTMENT OF THEORETICAL CHEMISTRY, THE UNIVERSITY OF SYDNEY.              C
C                                                                              C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      COMMON /B1/TTI,DELT,DELE,NREACT,FIN(20),RHO(300),AKI(300)
      COMMON /B2/ALI(20000),ALP(50),PW(50),ALF,NSTEP
      COMMON /B3/PROB(300),ANORM(300)
      COMMON /B4/EN(300),ENN(300),DELH,CP,SIG(300),BI(300)
      COMMON /B5/ALPHA,N,NI,IMAX,ITIME,LI
      COMMON /B8/NBAND
      COMMON /B9/OMG,CLA(300),CLS(300),AVY(300),AVB(300),F1(30000)
      EXTERNAL AUX
      EXTERNAL AUX1
      REAL KB
      DATA IKTMAX/800/,NIMAX/800/
      DATA TO/340./,KB/1.38/
      DIMENSION E(300),TITLE(20),GI(300),PRES(2),AKII(300),C2A(300)
      DIMENSION DGGI(300),A(9,300),AA(9,300),G(300,300),IP(300),B(300)
      DIMENSION F(300),FDP(1000),SIGAP(1),MTIME(1),IPLT(20),GKI(300)
      DIMENSION PLT(20)
      CALL ERRSET(208,100000,-1,1)
      IC=73
C READ DATA IN FROM UNIT 1 AND UNIT 30 (RRKM DATA SET)
      READ(30,1001)TITLE
      WRITE(3,1002)TITLE
      CALL SCAN(2,30,IC,FIN,9)
      IKT=FIN(8)
      WRITE(3,1005)
      IF(IKT.GT.IKTMAX)WRITE(3,1003)IKT,IKTMAX
      CALL SCAN(2,30,IC,AKII,IKT)
      CALL SCAN(1,30,IC,NI,1)
      IF(NI.GT.NIMAX)WRITE(3,1004)NI,NIMAX
      CALL SCAN(2,30,IC,RHO,NI)
      ICHD=FIN(9)
      CALL SCAN(1,1,IC,NPLT,1)
      CALL SCAN(2,1,IC,PLT,NPLT)
      CALL SCAN(2,1,IC,CONV,1)
      CALL SCAN(1,1,IC,ICPT,1)
      CALL SCAN(2,1,IC,ERROR,1)
      CALL SCAN(2,1,IC,ERMAX,1)
      CALL SCAN(2,1,IC,DELH,1)
      CALL SCAN(2,1,IC,ALPHA,1)
      CALL SCAN(2,1,IC,CP,1)
```

(Continued)

```
      CALL SCAN(2,1,IC,PRES,2)
      CALL SCAN(2,1,IC,SIGO,1)
      CALL SCAN(2,1,IC,XX,1)
      CALL SCAN(2,1,IC,BETA,1)
      CALL SCAN(2,1,IC,DELT,1)
      CALL SCAN(1,1,IC,ALF,1)
      CALL SCAN(2,1,IC,NSTEP,1)
      CALL SCAN(2,1,IC,ALP,NSTEP)
      CALL SCAN(2,1,IC,PW,NSTEP)
      DELH=DELH*1.E+3
C CALCULATE PARAMETERS AND CHECK RHO VECTOR FROM RRKM
      DO 136 K=1,NPLT
  136 IPLT(K)=PLT(K)*1.E-6/DELT
      LI=FIN(2)/FIN(1)
      DELE=FIN(1)
      EO=FIN(4)
      NREACT=EO*349.8/DELE
      IF(RHO(1).NE.0.)GO TO 100
      NI=NI-1
      DO 101 I=1,NI
  101 RHO(I)=RHO(I+1)
      NREACT=NREACT-1
      EO=EO-DELE*2.86E-3
C REARRANGE THE K(I) VECTOR TO CRITICAL ENERGY LEVEL
C AND INITIALISE THE GI,BI AND SIG VECTORS TO ZERO
  100 CONTINUE
      DO 102 I=1,NI
      IF(I.GT.NREACT)GO TO 103
      AKI(I)=0.
      GO TO 104
  103 AKI(I)=AKII(I-NREACT)
  104 GI(I)=0.
      SIG(I)=0.
      BI(I)=0.
  102 CONTINUE
      IT=1
      N=NI+1
      IMAX=NI-LI
      AMOLF=PRES(1)/(PRES(1)+PRES(2))
      C1=(PRES(2))*4.41313E+7*(FIN(5)**2.)
      C2=FIN(6)*FIN(7)/(FIN(6)+FIN(7))
      OMG1=C1/SQRT(TO*C2)
      EOC=FIN(4)*4.1812
      PR1=PRES(1)*133.33
      PR2=PRES(2)*133.33
      WRITE(3,1006)
      WRITE(3,1007)FIN(6),FIN(7),FIN(5)
      WRITE(3,1005)
      WRITE(3,1008)
      WRITE(3,1009)PR1,PRES(1),PR2,PRES(2),OMG1
      WRITE(3,1005)
```

(Continued)

```
      WRITE(3,1010)
      WRITE(3,1011)ALF,FIN(3),FIN(1)
      WRITE(3,1005)
      WRITE(3,1012)
      WRITE(3,1013)NSTEP,DELT,ICHD
      WRITE(3,1005)
      WRITE(3,1014)
      WRITE(3,1015)IKT,NI,ERROR
      WRITE(3,1005)
      WRITE(3,1016)
      WRITE(3,1017)DELH,ALPHA,CP
      WRITE(3,1005)
      WRITE(3,1018)
      WRITE(3,1005)
      WRITE(3,1019)
      WRITE(3,1020)SIGO,XX,BETA
      WRITE(3,1005)
      WRITE(3,1025)
      WRITE(3,1026)FIN(4),ECC
      WRITE(3,1005)
C SET INITIAL DISTRIBUTION TO THERMAL BOLTZMANN DISTRIBUTION
C AND CALCULATE CROSS SECTION PROFILE VECTOR SIG(I)
      SUMG=0.
      DO 105 J=1,NI
      EN(J)=FLOAT(J)*DELE
      ENN(J)=EN(J)*1.986/KB
      NPHT=J/LI
      SIG(J)=SIGO*FEXP(-XX*(NPHT)**BETA)
      GI(J)=FEXP(ALOG(RHO(J))-ENN(J)/TC)
      C2A(J)=SIG(J)/(FIN(2)*1.986E-23)
  105 SUMG=SUMG+GI(J)
      SMAX=SIGO
      ENMAX=EN(NI)
C PLOT ABSORPTION CROSS SECTION
      CALL XYPAPE(-8.,-8.,0.,ENMAX,0.,SMAX)
      CALL XYHEAD(TITLE)
      CALL XYTLEN('$')
      CALL XYNAMX('ENERGY LEVEL (CM-1)$')
      CALL XYNAMY('ABSORPTION CROSS-SECTION (CM@22@)$')
      CALL XYLINE(EN,SIG,NI)
      WRITE(3,1033)
      WRITE(3,1034)(SIG(J),J=1,NI)
      GMAX=0.
      DO 106 J=1,NI
      GI(J)=GI(J)/SUMG
      IF(GI(J).GT.GMAX)GMAX=GI(J)
      E(J)=AMAX1(ERROR*GI(J),ERMAX)
  106 CONTINUE
      GI(N)=TO
      E(N)=TO*ERROR
      WRITE(3,1035)GI(N)
```

(Continued)

```
      WRITE(3,1034)(GI(J),J=1,NI)
      WRITE(3,1036)
      WRITE(3,1034)(RHO(J),J=1,NI)
      WRITE(3,1037)
      WRITE(3,1034)(AKI(J),J=1,NI)
C CALL SUBROUTINE LASERP TO CALCULATE LASER POWER TIME PROFILE
      CALL LASERP
      NTIME=TTI/DELT
      SUMT=0.
      IFAIL=0.
      TIME=0.
      HO=DELT
      H=HO/4.
      IWRITE=1
      WRITE(3,1038)NTIME
      VAX=EN(NI)/10.
      VAY=0.18
      XMAX=EN(NI)
C PLOT THE DISTRIBUTION AT TIME ZERO
      CALL XYPAPE(-16.,-7.5,0.,XMAX,0.,.2)
      CALL XYHEAD(TITLE)
      CALL XYNUMB(VAX,VAY,PRES(1),'(''PRESSURE OF REACTANT'',G10.3,''TOR4
     1R'')',0.07,0.)
      CALL XYNUMB(VAX,VAY-.02,PRES(2),'(''PRESSURE OF BUFFER '',G10.3,''TO
     1TORR'')',0.07,0.)
      CALL XYNUMB(VAX,VAY-.04,ALF,'(''LASER FLUENCE'',G10.3,''J/CM2'')',0
     10.07,0.)
      CALL XYTLEN('$')
      CALL XYNAMX('ENERGY (CM@2-1@)$')
      CALL XYNAMY('POPULATION$')
      CALL XYLINE(EN,GI,NI)
      WRITE(3,1039)
      SUMC=0.
      SUML=0.
      DO 107 ITIME=1,NTIME
      SUMB=0.
      TEMP=GI(N)
      OMG=C1/SQRT(TEMP*C2)
C CALCULATE ABSORPTION AND EMISSION RATES LI(I) AND LS(I)
      DO 108 I=1,NI
      CLA(I)=ALI(ITIME)*C2A(I)
      CLS(I)=0.
      IF(I.GT.LI)CLS(I)=CLA(I-LI)*RHO(I-LI)/RHO(I)
      BI(I)=FEXP(ALOG(RHO(I))-ENN(I)/TEMP)
  108 SUMB=SUMB+BI(I)
      DO 109 I=1,NI
  109 BI(I)=BI(I)/SUMB
CALL SUBROUTINE ENEXPT TO CALCULATE THE TRANSITION PROBABILITY
C VECTOR PROB(I) AND NORMALISATION FACTOR ANORM(I)
      CALL ENEXPT
      IJ=0
```

(Continued)

```
      DO 110 I=1,NI
      AVY(I)=GI(I)*AMOLF
      AVB(I)=BI(I)*AMOLF
      JMIN=MAXO(1,I-NBAND+1)
      IF(JMIN.EQ.I)GO TO 111
      DO 112 J=JMIN,I
      IJ=IJ+1
      F1(IJ)=FEXP((ALOG(RHO(I))-ALOG(RHO(J)))+((ENN(J)-ENN(I))/TEMP))
  112 CONTINUE
  111 JMAX=MINO(NI,I+NBAND-1)
      IF(JMAX.EQ.I)GO TO 113
      DO 114 J=I,JMAX
      IJ=IJ+1
      F1(IJ)=FEXP((ALOG(RHO(J))-ALOG(RHO(I)))+((ENN(I)-ENN(J))/TEMP))
  114 CONTINUE
  113 CONTINUE
  110 CONTINUE
C CALL SUBROUTINE GEAR TO SOLVE THE N COUPLED DE'S
      CALL GEAR(TIME,GI,E,IT,N,HO,H,AUX,A,AA,G,IP,B,F,K,IFAIL)
      DO 115 I=1,NI
      IF(GI(I).LT.0.)GI(I)=0.
  115 CONTINUE
C WRITE THE G(I) VECTOR EACH IOPT ITERATION
      IF(ICPT.EQ.0)GO TO 116
C     CALL XYLINE(EN,GI,NI)
      IWR=ITIME/(IOPT*IWRITE)
      IF(IWR.NE.1)GO TO 116
      WRITE(3,1040)ITIME
      WRITE(3,1034)(GI(I),I=1,NI)
      WRITE(3,1039)
      IWRITE=IWRITE+1
      GO TO 117
  116 CONTINUE
  117 SGT=0.
C RE-CALCULATE THE ERROR VECTOR AND CALCULATE THE FRACTION
C DISSOCIATED AND CALCULATE THE APPARENT CROSS-SECTION
      GKIMAX=0.
      DO 118 I=1,NI
      E(I)=AMAX1(ERROR*GI(I),ERMAX)
      SUMC=SUMC+(GI(I)*SIG(I)*ALI(ITIME)*DELT)
      GKI(I)=GI(I)*AKI(I)
      IF(GKI(I).GT.GKIMAX)GKIMAX=GKI(I)
  118 SGT=SGT+GI(I)
      FD=1.-SGT
      SUML=0.
      DO 119 I=1,ITIME
  119 SUML=SUML+ALI(I)*DELT
      SIGAP(1)=SUMC/SUML
      ALFD=0.
      IF(FD.GT.0.)ALFD=ALOG(FD)
      SSZ=0.
```

(Continued)

```
      SS1=0.
      SS2=0.
      GIMAX=0.
      DO 120 I=1,NI
      GKI(I)=GKI(I)/(GKIMAX*20.)
      IF(GI(I).GT.GIMAX)GIMAX=GI(I)
      SSZ=SSZ+GI(I)
      FLT=FLOAT(I)
      SS1=SS1+FLT*GI(I)
  120 SS2=SS2+FLT*FLT*GI(I)
      BARI=SS1/SSZ
      SDI=(SS2/SSZ-SS1**2./SSZ**2.)**.5
      WRITE(3,1021)ITIME,TIME,GI(N),FD,ALFD,SUML,SIGAP(1),BARI,SDI
C TEST FOR PLOT REQUEST
      DO 121 I=1,NPLT
      IPLTT=IPLT(I)
      IF(ITIME.NE.IPLTT)GO TO 121
      VX=BARI*DELE/LI
      SDP=SDI*DELE/LI
      PLTT=PLT(I)
      CALL XYNUMB(VX,GIMAX+.02,VX,'(''MEAN'',F7.2,''(CM-1)'')',.07,0.)
      CALL XYNUMB(VX,GIMAX+.01,SDP,'(''SD'',F8.3,''(CM-1)'')',.07,.0)
      CALL XYNUMB(VX,GIMAX+.03,PLTT,'(''TIME'',F4.1,''10-6 SECS'')',.07,
     1.0)
      CALL XYSETL(1)
      CALL XYLINE(EN,GI,NI)
      CALL XYSETL(2)
      CALL XYLINE(EN,GKI,NI)
      GO TO 107
  121 CONTINUE
  107 CONTINUE
      WRITE(3,1022)
      IK=1
      FDP(IK)=FD
  135 SUMB=0.
      TEMP=GI(N)
      OMG=C1/SQRT(TEMP*C2)
      DO 122 I=1,NI
      BI(I)=FEXP(ALOG(RHO(I))-ENN(I)/TEMP)
  122 SUMB=SUMB+BI(I)
      DO 123 I=1,NI
  123 BI(I)=BI(I)/SUMB
      CALL ENEXPT
      IJ=0
      DO 124 I=1,NI
      AVY(I)=GI(I)*AMOLF
      AVB(I)=BI(I)*AMOLF
      JMIN=MAX0(1,I-NBAND+1)
      IF(JMIN.EQ.I)GO TO 125
      DO 126 J=JMIN,I
      IJ=IJ+1
```

(Continued)

```
      F1(IJ)=FEXP((ALOG(RHO(I))-ALOG(RHO(J)))+((ENN(J)-ENN(I))/TEMP))
  126 CONTINUE
  125 JMAX=MIN0(NI,I+NBAND-1)
      IF(JMAX.EQ.I)GO TO 127
      DO 128 J=I,JMAX
      IJ=IJ+1
      F1(IJ)=FEXP((ALOG(RHO(J))-ALOG(RHO(I)))+((ENN(I)-ENN(J))/TEMP))
  128 CONTINUE
  127 CONTINUE
  124 CONTINUE
      CALL GEAR(TIME,GI,E,IT,N,HO,H,AUX1,A,AA,G,IP,B,F,K,IFAIL)
      DO 129 I=1,NI
      IF(GI(I).LT.0.)GI(I)=0.
  129 CONTINUE
      ITIME=ITIME+1
C WRITE THE G(I) VECTOR EACH IOPT ITERATION
      IF(ICPT.EQ.0)GO TO 130
      IWR=ITIME/(IOPT*IWRITE)
      IF(IWR.NE.1)GO TO 130
      WRITE(3,1040)ITIME
      WRITE(3,1034)(GI(I),I=1,NI)
      WRITE(3,1023)
      IWRITE=IWRITE+1
      GO TO 131
  130 CONTINUE
  131 SGT=0.
C RE-CALCULATE THE ERROR VECTOR AND CALCULATE THE FRACTION DISSOCIATED
      GKIMAX=0.
      DO 132 J=1,NI
      E(J)=AMAX1(ERROR*GI(J),ERMAX)
      GKI(J)=GI(J)*AKI(J)
      IF(GKI(J).GT.GKIMAX)GKIMAX=GKI(J)
  132 SGT=SGT+GI(J)
      IK=IK+1
      FDP(IK)=1.-SGT
      ALFDP=0.
      IF(FDP(IK).GT.0.)ALFDP=ALOG(FDP(IK))
      SSZ=0.
      SS1=0.
      SS2=0.
      GIMAX=0.
      DO 133 I=1,NI
      GKI(I)=GKI(I)/(GKIMAX*20.)
      IF(GI(I).GT.GIMAX)GIMAX=GI(I)
      SSZ=SSZ+GI(I)
      FLT=FLOAT(I)
      SS1=SS1+FLT*GI(I)
  133 SS2=SS2+FLT*FLT*GI(I)
      BARI=SS1/SSZ
      SDI=(SS2/SSZ-SS1**2./SSZ**2.)**.5
      WRITE(3,1024)ITIME,TIME,GI(N),FDP(IK),ALFDP,BARI,SDI
```

(Continued)

```
C TEST FOR PLOT REQUEST
      DO 134 J=1,NPLT
      ITEST=NTIME+IK
      IPLTT=IPLT(J)
      IF(ITEST.NE.IPLTT)GO TO 134
      XMN=(BARI-50.)*CELE
      XMX=(BARI+50.)*CELE
      VX=BARI*CELE
      PLTT=PLT(J)
      CALL XYPAPE(-8.,-6.,XMN,XMX,0.,.06)
      CALL XYHEAD(TITLE)
      CALL XYNAMX('ENERGY (CM@2-1@$')
      CALL XYNAMY('POPULATION$')
      CALL XYNUMB(VX,GIMAX+.2,BARI,'(''MEAN'',F7.2,''(CM-1)'')',.07,.0)
      CALL XYNUMB(VX,GIMAX+.1,SDI,'(''SD'',F8.3,''(CM-1)'')',.07,.0)
      CALL XYNUMB(VX,GIMAX+.3,PLTT,'(''TIME'',F4.1,''10-6 SECS'')',.07,0
     1.)
      CALL XYSETL(1)
      CALL XYLINE(EN,GI,NI)
      CALL XYSETL(2)
      CALL XYLINE(EN,GKI,NI)
      GO TO 137
  134 CONTINUE
  137 CONTINUE
      DIFF=FDP(IK)-FDP(IK-1)
      IF(DIFF.GT.CONV*FDP(IK-1).AND.TIME.LT.10.E-6)GO TO 135
      CALL XYEND
 1001 FORMAT(20A4)
 1002 FORMAT(1X,20A4,/)
 1003 FORMAT(5X,'INDEX IKT(',I6,')HAS EXCEEDED THE DIMENSION IKTMAX(',I6
     1,')')
 1004 FORMAT(5X,'INDEX NI(',I6,') HAS EXCEEDED THE DIMENSION NIMAX(',I6,'
     1')')
 1005 FORMAT(1X,':::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     1:::::::::::::::::::')
 1006 FORMAT(1X,':',4X,'MASS OF REACTANT',4X,':',5X,'MASS OF BUFFER',5X,,
     1':',3X,'COLLISION DIAMETER',3X,':',/,1X,':',9X,'(AMU)',10X,':',9X,
     2'(AMU)',10X,':',10X,'(A)',11X,':')
 1007 FORMAT(1X,':',8X,F7.2,9X,':',8X,F7.2,9X,':',8X,F7.2,9X':')
 1008 FORMAT(1X,':',2X,'PRESSURE OF REACTANT',2X,':',3X,'PRESSURE OF BUF
     1FFER',3X,':',9X,'OMEGA',10X,':',/,1X,':',2X,'(PA)',9X,'(TORR)',3X,':'
     2:',2X,'(PA)',9X,'(TORR)',3X,':',8X,'(SEC-1)',9X,':')
 1009 FORMAT(1X,':',2X,F5.0,8X,F7.3,2X,':',2X,F5.0,8X,F7.3,2X,':',5X,1PG
     113.6,6X,':')
 1010 FORMAT(1X,':',5X,'LASER FLUENCE',6X,':',4X,'LASER FREQUENCY',5X,':
     1',7X,'GRAIN SIZE',7X,':',/,1X,':',8X,'(J CM-2)',8X,':',9X,'(CM-1)'
     2,9X,':',9X,'(CM-1)',9X,':')
 1011 FORMAT(1X,':',8X,F8.3,8X,':',8X,F8.3,8X,':',8X,F8.3,8X,':')
 1012 FORMAT(1X,':',3X,'NUMBER LASER STEPS',3X,':',6X,'TIME STEP',9X,':'
     1,3X,'NUMBER OF CHANNELS',3X,':',/,1X,':',24X,':',9X,'(SEC)',10X,':
     2',24X,':')
```

(Continued)

```
 1013 FORMAT(1X,':',10X,I4,10X,':',7X,1PG10.3,7X,':',10X,I4,10X,':')
 1014 FORMAT(1X,':',5X,'NUMBER OF K(I)',5X,':',4X,'NUMBER OF RHO(I)',4X,
     1':',3X,'G(I) ERROR FACTOR',4X,':')
 1015 FORMAT(1X,':',10X,I4,10X,':',10X,I4,10X,':',8X,F7.5,9X,':')
 1016 FORMAT(1X,':',4X,'ENTHALPY CHANGE',5X,':',7X,'<E(I+1,I)>',7X,':',5
     1X,'SPECIFIC HEAT',6X,':',/,1X,':',7X,'(CAL/MOL)',8X,':',9X,'(CM-1)
     2',9X,':',6X,'(CAL/MOL/K)',7X,':')
 1017 FORMAT(1X,':',7X,F10.3,7X,':',7X,F10.3,7X,':',7X,F10.3,7X,':')
 1018 FORMAT(1X,':',2X,'ABSORPTION CROSS SECTION :: SIG(I)=SIG(0)*EXP(-X
     1*I**BETA)',15X,':')
 1019 FORMAT(1X,':',8X,'SIGMA(0)',8X,':',11X,'X',12X,':',10X,'BETA',10X,
     1':',/,1X,':',9X,'(CM2)',10X,':',24X,':',24X,':')
 1020 FORMAT(1X,':',5X,1PG13.6,6X,':',5X,G13.6,6X,':',5X,G13.6,6X,':')
 1033 FORMAT(1X,'CROSS SECTION VECTOR')
 1034 FORMAT(1X,10E10.3)
 1035 FORMAT(1X,'DISTRIBUTION AT TIME ',E AND TEMPERATURE ',F10.2,'K')
 1036 FORMAT(1X,'DENSITY OF STATES, RHO(I)')
 1037 FORMAT(1X,'RATE CONSTANTS, K(I)')
 1038 FORMAT(10X,'NUMBER OF TIME STEPS=',I10)
 1039 FORMAT(2X,'STEP',1X,'TIME (SEC)',1X,'BUFFER T (K)',9X,'F',6X,'LOG
     1 F',5X,'FLUENCE',8X,'EXP SIG',3X,'MEAN I',2X,'   SD')
 1040 FORMAT(1X,'LISTING OF GI VECTOR ',I10,' ITERATION')
 1021 FORMAT(1X,I4,1X,1PG10.3,2X,G10.3,5X,G10.3,2X,G10.3,3X,0PF5.3,6X,1P
     1G10.3,1X,0PF7.2,1X,F8.3)
 1022 FORMAT(1X,'**********************************************************
     1',/,1X,'LASER TURNED OFF AND PROCEED',/,'**************************
     2**********************************',/)
 1023 FORMAT(1X,'STEP',1X,'TIME (SEC)',1X,'BUFFER T (K)',9X,'F',6X,'LOG
     1 F',3X,'MEAN I',2X,'   SD')
 1024 FORMAT(1X,I4,1X,1PG13.3,2X,G10.3,5X,G10.3,2X,G10.3,1X,0PF7.2,1X,F8
     1.3)
 1025 FORMAT(1X,':',4X,'CRITICAL ENERGY',5X,':',24X,':',24X,':',/,1X,':'
     1,2X,'(KCAL/MOL)',2X,'(KJ/MOL)',2X,':',24X,':',24X,':')
 1026 FORMAT(1X,':',2X,F9.3,2X,F9.3,2X,':',24X,':',24X,':')
      STOP
      END
      SUBROUTINE AUX(X,Y,F)
C
C SUBROUTINE AUX SETS UP THE COUPLED DE'S FOR SUBROUTINE GEAR
C THIS ROUTINE CONTAINS THE BASIC THEORETICAL EQUATIONS
C
      COMMON /B1/TTI,DELT,DELE,NREACT,FIN(20),RHO(300),AKI(300)
      COMMON /B2/ALI(2000),ALP(50),PW(50),ALF,NSTEP
      COMMON /B3/PROB(300),ANORM(300)
      COMMON /B4/EN(300),ENN(300),DELH,CP,SIG(300),BI(300)
      COMMON /B5/ALPHA,N,NI,IMAX,ITIME,LI
      COMMON /B8/NBAND
      COMMON /B9/OMG,CLA(300),CLS(300),AVY(300),AVB(300),F1(30000)
      DIMENSION Y(300),OGBI(300),F(300)
      SUM4=0.
      IJ=0
```

(Continued)

```
      DO 5 I=1,NI
      ANORMI=ANORM(I)
      YI=Y(I)
      ENI=EN(I)
      ENNI=ENN(I)
      AVYI=AVY(I)
      SUM1=0.
      SUM2=0.
      SUM3=0.
      JMIN=MAXO(1,I-NBAND+1)
      IF(JMIN.EQ.I)GO TO 44
      DO 2 J=JMIN,I
      IJ=IJ+1
      F1IJ=F1(IJ)
      S1=PROB(I-J+1)*ANORMI*(Y(J)*F1IJ-YI)
      S3=PROB(I-J+1)*ANORMI*(AVY(J)*F1IJ-AVYI)*(EN(J)-ENI)
      IF(J.NE.JMIN.AND.J.NE.I)GO TO 6
      S1=S1*0.5
      S3=S3*0.5
    6 SUM3=SUM3+S3
    2 SUM1=SUM1+S1
   44 JMAX=MINO(NI,I+NBAND-1)
      IF(JMAX.EQ.I)GO TO 55
      DO 3 J=I,JMAX
      IJ=IJ+1
      S2=PROB(J-I+1)*ANORM(J)*(Y(J)-YI*F1(IJ))
      IF(J.NE.I.AND.J.NE.JMAX) GO TO 4
      S2=S2*0.5
    4 CONTINUE
    3 SUM2=SUM2+S2
   55 SUM=SUM1+SUM2
      TERM1=OMG*DELE*SUM
      IF(I.LE.LI.OR.I.GT.IMAX)GO TO 11
      TERM2=-(AKI(I)+CLA(I)+CLS(I))*YI+CLA(I-LI)*Y(I-LI)+CLS(I+LI)*Y(I+L
     1I)
      GO TO 12
   11 IF(I.GT.IMAX)GO TO 13
      TERM2=-(AKI(I)+CLA(I)+CLS(I))*YI+CLS(I+LI)*Y(I+LI)
      GO TO 12
   13 TERM2=-(AKI(I)+CLA(I)+CLS(I))*YI+CLA(I-LI)*Y(I-LI)
   12 TERM3=(2.86*EN(I)-DELH)*AKI(I)*AVY(I)
      TERM4=2.*OMG*DELE*SUM3*2.859
      SUM4=SUM4+TERM3+TERM4
    5 F(I)=TERM1+TERM2
      F(N)=SUM4/CP
      RETURN
      END
      SUBROUTINE AUX1(X,Y,F)
C
C SUBROUTINE AUX1 SETS UP THE COUPLED DE'S FOR SUBROUTINE GEAR
C THIS ROUTINE CONTAINS THE BASIC THEORETICAL EQUATIONS
```

(Continued)

```
C FOR CONDITIONS AFTER THE LASER PULSE
C
      COMMON /B1/TTI,DELT,DELE,NREACT,FIN(20),RHO(300),AKI(300)
      COMMON /B2/ALI(20000),ALP(50),PW(50),ALF,NSTEP
      COMMON /B3/PROB(300),ANORM(300)
      COMMON /B4/EN(300),ENN(300),DELH,CP,SIG(300),BI(300)
      COMMON /B5/ALPHA,N,NI,IMAX,ITIME,LI
      COMMON /B8/NBAND
      COMMON /B9/OMG,CLA(300),CLS(300),AVY(300),AVB(300),F1(30000)
      DIMENSION Y(300),OGBI(300),F(300)
      SUM4=0.
      IJ=0
      DO 5 I=1,NI
      ANORMI=ANORM(I)
      YI=Y(I)
      ENI=EN(I)
      ENNI=ENN(I)
      AVYI=AVY(I)
      SUM1=0.
      SUM2=0.
      SUM3=0.
      JMIN=MAX0(1,I-NBAND+1)
      IF(JMIN.EQ.I)GO TO 44
      DO 2 J=JMIN,I
      IJ=IJ+1
      F1IJ=F1(IJ)
      S1=PROB(I-J+1)*ANORMI*(Y(J)*F1IJ-YI)
      S3=PROB(I-J+1)*ANORMI*(AVY(J)*F1IJ-AVYI)*(EN(J)-ENI)
      IF(J.NE.JMIN.AND.J.NE.I)GO TO 6
      S1=S1*0.5
      S3=S3*0.5
    6 SUM3=SUM3+S3
    2 SUM1=SUM1+S1
   44 JMAX=MIN0(NI,I+NBAND-1)
      IF(JMAX.EQ.I)GO TO 55
      DO 3 J=I,JMAX
      IJ=IJ+1
      S2=PROB(J-I+1)*ANORM(J)*(Y(J)-YI*F1(IJ))
      IF(J.NE.I.AND.J.NE.JMAX) GO TO 4
      S2=S2*0.5
    4 CONTINUE
    3 SUM2=SUM2+S2
   55 SUM=SUM1+SUM2
      TERM1=OMG*DELE*SUM
      TERM2=-AKI(I)*YI
      TERM3=(2.86*EN(I)-DELH)*AKI(I)*AVY(I)
      TERM4=2.*OMG*DELE*SUM3*2.859
      SUM4=SUM4+TERM3+TERM4
    5 F(I)=TERM1+TERM2
      F(N)=SUM4/CP
      RETURN
```

(Continued)

```
      END
      FUNCTION FEXP(EX)
      FEXP=0.
      IF(ABS(EX).LT.150.)FEXP=EXP(EX)
      RETURN
      END
      SUBROUTINE LASERP
C
C SUBROUTINE LASERP CALCULATES THE LASER POWER ALI(TIME) FROM
C THE INPUT STEP FUNCTION
C
      COMMON /B1/TTI,DELT,DELE,NREACT,FIN(20),RHO(300),AKI(300)
      COMMON /B2/ALI(2000),ALP(50),PW(50),ALF,NSTEP
      DIMENSION NCH(50)
      NCH(1)=PW(1)/DELT
      TTI=PW(1)
      I=0
      J=0
      DO 1 K=2,NSTEP
      NCH(K)=NCH(K-1)+PW(K)/DELT
    1 TTI=TTI+PW(K)
      SUM=0.
      DO 2 M=1,NSTEP
    2 SUM=SUM+ALP(M)*PW(M)
    5 I=I+1
    4 J=J+1
      ALI(J)=ALP(I)*ALF/SUM
      IF(J.LT.NCH(I))GO TO 4
      IF(I.LT.NSTEP)GO TO 5
      JMAX=NCH(NSTEP)
      RETURN
      END
```

(Continued)

```
      SUBROUTINE ENEXPT
C  COMPUTES COLLISION PROBABILITY ARRAY PROB.
C  TO CHANGE THE PROBABILITY FUNCTION,(1) CHANGE FORMAT STATEMENT;
C  (2)  CHANGE FUNCTIONAL FORM
      COMMON /B3/PROB(300),ANORM(300)
      COMMON /B1/TTI,DELT,DELE,NREACT,FIN(20),RHO(300),AKI(300)
      COMMON /B4/EN(300),ENN(300),DELH,CP,SIG(300),BI(300)
      COMMON /B5/ALPHA,N,NI,IMAX,ITIME,LI
      COMMON /B8/NBAND
      NBAND=0
      DO 809 I=1,NI
      PROB(I)=0.
  809  CONTINUE
C  GENERATE UN-NORMALIZED PROBABILITY MATRIX.
      DO 1 I=1,NI
      DE=FLOAT(I-1)*DELE
      X=DE/ALPHA
      IF(X.LT.10.) PROB(I)=EXP(-X)
      IF(X.GE.10.)GO TO 2
  1    CONTINUE
  2   CONTINUE
C  GENERATE ANORM, THE VECTOR OF NORMALIZERS,INITIALLY AS C,ITS RECIPROCAL.
      NBAND=I-1
      IF(NBAND.EQ.1) NBAND=0
      IF(NBAND.EQ.0) RETURN
COMMENCE FINITE DIFFERENCE SOLUTION OF INTEGRAL EQUATION FOR C.
      J=NI+1
      DO 5 JJ=1,NI
      J=J-1
      IF(J.GE.(NREACT)) GO TO 4
C  TO AVOID OCCASIONAL PROBLEMS WITH NORMALIZATION ALGORITHM FOR STATES
C  WITH VERY LOW ENERGIES, ALL STATES WITH ENERGIES BELOW EC/2 ARE GIVEN THE
C  SAME NORMALIZATION .  THIS HAS
C  NO PHYSICAL EFFECT,SINCE THESE ALWAYS HAVE THEIR EQUILIBIRIUM POPULATIONS
      ANORM(J)=ANORM(J+1)
      GO TO 5
  4   CONTINUE
      ANORM(J)=0.
      IF(J.EQ.1) GO TO 3
      DE=0.
      IMIN=MAX0(1,J-NBAND+1)
      DO 7 I=IMIN,J
      A=PROB(J-I+1)
      IF(I.EQ.J.OR.I.EQ.IMIN) A=A*0.5
      DE=DE+A
  7   CONTINUE
      ANORM(J)=DE*DELE
  3   CONTINUE
      IF(J.EQ.NI) GO TO 5
```

(Continued)

```
      IUP=MINC(NI,J+NBAND-1)
      DE=0.
      IF(IUP.LE.J) GC TO 5
      DO 6 I=J,IUP
      A=(PROB(I-J+1)*BI(I)/ANORM(I))
      IF(I.EQ.J.OR.I.EQ.IUP) A=A*0.5
      DE=DE+A
    6 CONTINUE
      IF(BI(J).EQ.C.)GO TO 20
      ANORM(J)=ANORM(J)/(1.-(DELE*DE/BI(J)))
      GO TC 5
   20 ANORM(J)=ANGRM(J)
    5 CONTINUE
      DC 13 J=1,NI
      IF(ANORM(J).EQ.C.)GO TO 21
       ANORM(J)=1./ANCRM(J)
      GO TC 13
   21 ANORM(J)=1./ALPHA
   13 CONTINUE
      RETURN
      END
```

APPENDIX C
SUBROUTINE GEAR (NAGFLIB : 793/504 : Mk 5 : Nov 74)

C1. PURPOSE

GEAR integrates a system of first order ordinary differential equations over a range [Gear 1971].

C2. SPECIFICATION (FORTRAN IV)

```
SUBROUTINE GEAR (X,Y,E,T,N,HO,H,AUX,A,AA,G,IP,B,F,K,IFAIL)
INTEGER  T,N,IP,K,IFAIL
REAL  X,Y,E,HO,H,A,AA,G,B,F
DIMENSION  Y(N),E(N),A(9,N),G(N,N),B(N),F(N),AA(9,N)
EXTERNAL AUX.
```

C3. DESCRIPTION

GEAR integrates a system of ordinary differential equations

$$\frac{dy_i}{dx} = f_i(x,y_1,y_2,\ldots,y_n), i = 1,\ldots,r$$

from X to X+HO. The method is based on ideas given by Gear [1971], and is primarily intended for 'stiff' systems of equations. Most stiff systems can be characterised as having complementary functions (transient) which decay more rapidly than the free solution $y_1,y_2,\ldots,y_n$. In most cases we can quantify the degree of stiffness of the system of differential equations by the size of the stiffness ratio (the larger the ratio, the stiffer the system). If the eigenvalues of the Jacobian matrix $|\partial f_i/\partial y_j|$ are $\lambda_1,\lambda_2,\ldots,\lambda_n$, then the stiffness ratio is defined by max. $(Re(\lambda_i))$/min. $(Re(\lambda_i))$. Note that, in general, the values λ_i will depend on x and on the solution $y_1,y_2,\ldots,y_n$ and hence the degree of stiffness of a problem may vary across the range. The subroutine chooses the step length h and the order of the method so that the specified accuracy is obtained with minimum computation. The methods used vary in order of accuracy, with a maximum order of six (that is, the local error is of order h^6). The differential equations are defined by a subroutine AUX which evaluates the derivatives f_i in terms of x and $y_1,y_2,\ldots,y_n$.

C4. PARAMETERS

X - real

Before entry, X must be set to the initial value of the independent variable x, and on exit it will contain X+HO, unless an error has occurred, in which case it will contain the last calculated value of the independent variable. Not to be disturbed between steps if the integration is to continue.

Y - real array of dimension (N)

Before entry, Y must be set to the values of $y_1, y_2, \ldots, y_N$ at X, and on exit it will contain the computed values at X+HO, unless an error has occurred in which case it will contain the last calculated values of $y_1, y_2, \ldots, y_N$. Not to be disturbed between steps if the integration is to continue.

E - real array of dimension (N).

Before entry, E must be set to the error bounds for each component of the solution. The type of error test required (relative, absolute or mixed) is specified by the parameter T. Unchanged on exit.

T - integer.

Before entry, T must be set to a value, the modulus of which defines the error test to be applied. If the estimate of the local error is TR(I), then

$|T| = 1$ gives a mixed error test : $|TR(I)| < E(I)\ (1+|Y(I)|)$,

$|T| = 2$ gives an absolute error test : $|TR(I)| < E(I)$,

$|T| = 3$ gives a relative error test : $|TR(I)| < E(I)\ |Y(I)|$.

This is used to determine the step-length and order of the method. For most cases T=1 is satisfactory. The sign of T is used in the subroutine to distinguish the first entry. T should be positive on the first entry and should not be altered between steps when integration is to be continued. On first exit, the sign of T will be changed but thereafter it will remain negative.

N - integer

On entry, N specifies the number of equations. Not to be disturbed between steps if the integration is to continue. Unchanged on exit.

HO - real

On entry, HO specifies the interval over which integration is required. Unchanged on exit.

H - real

Before entry, H must be set to an estimate of the step-length needed for integration, although the subroutine will modify this if necessary to maintain local accuracy. If H is zero on entry it will be set initially to HO/4. On exit, H will contain the current step-length. Not to be disturbed between steps if the integration is to continue.

AUX - subroutine, supplied by the user, with specification:

SUBROUTINE AUX(X,Y,F)
real F(n),Y(n),X
where n is the number of equations being solved. It evaluates the derivatives of Y(1),Y(2),...,Y(N) at a general point X and places them in F(1),F(2),...,F(N). AUX must be declared as EXTERNAL in the (sub) program from which GEAR is called.

A - real array of dimension (9,N)

Used as working space. Not to be disturbed between steps if the integration is to continue.

AA - real array of dimension (9,N)

Used as working space. Not to be disturbed between steps if the integration is to continue.

G - real array of dimension (N,N)

Used as working space. Not to be disturbed between steps if the integration is to continue.

IP - integer array of dimension (N)

Used as working space. Not to be disturbed between steps if the integration is to continue.

B - real array of dimension (N)

Used as working space. Not to be disturbed between steps if the integration is to continue.

F - real array of dimension (N)

Used as working space. Not to be disturbed between steps if the integration is to continue.

K - integer

Used for working space - contains the current order. Not to be disturbed between steps if the integration is to continue.

IFAIL - integer

Before entry, IFAIL must be assigned a value. For users not familiar with this parameter the recommended value is 0. Unless the routine detects an error, IFAIL contains 0 on exit.

C5. ERROR INDICATORS

Errors detected by the routine:

IFAIL = 1 This indicates that the step-length has been halved repeatedly until it is less than 10^{-4}(initial step-length); or too many steps are required to reach the end of the range.

IFAIL = 2 This indicates that the subroutine is unable to perform the corrector iterations, probably because of an unfortunate choice of H.

If the program fails with IFAIL = 1 or IFAIL = 2 then, as a first strategy, the subroutine should be called again with a smaller value of H starting with the last reliable values of the solution. If the program fails continually then it is likely that the solution of the differential system is ill-behaved (e.g. it is discontinuous or singular).