Eindhoven University of Technology

MASTER

Beam experiments on KrF, XeF, KrBr and XeBr excimer formation

Vrakking, Marc

Award date:
1987

COMPUTER PROGRAMS FOR THE SIMULATION
OF BOUND-FREE EMISSION SPECTRA

Marc Vrakking VDF/NO 87-19

Appendix belonging to EUT Internal Report VDF/NO 87-17

INHOUDSOPGAVE

EXPLANATION OF THE PURPOSE OF THE MAJOR SOURCE PROGRAMS

LISTINGS OF THE MAJOR SOURCE PROGRAMS

COMPUTER PROGRAMS FOR THE SIMULATION OF BOUND-FREE EMISSION SPECTRA

In order to analyse experimental bound-free rare gas halide emission spectra a number of computer-programs have been written on the central Burroughs 7900 mainframe at the Eindhoven University of Technology. In this report a summary will be given of the purpose of the major source-programs.

The total program structure is shown in Figure 1. Names starting with an 'S' are source-files containing programs written in FORTRAN (except programs **S/CONVERTER** and **S/CONVERTER/MAT**). Names starting with a 'D' are data-files. All files are stored under user TNNDAW20, where the programs are stored as **MINIB2/**'S' or 'D'**/GEBONDENVRIJ/FILENAME**.

Program S/POTCALC

Program **S/POTCALC** is largely based on a program which was already written at IMS[1]. In this program the emission-spectra of individual vibrational levels are determined by calculating the overlap-integral $\langle v|\mu_{eg}|E_g\rangle$ of the radial wave-functions of the bound vibrational upper states $|v\rangle$ and the free lower states $|E_g\rangle$.

For the calculation of the vibrational states $|v\rangle$ a method is applied, which was introduced by Cooley[2,3]. This is essentially the application of 6th order Numerov-integration in a predictor-corrector method. The free lower state wave-functions are calculated by a simple outward integration. According to the WKB-method these wave-functions are normalized, from conservation of particles ($|\psi|^2 \sim E^{-1/2}$), by the condition that the amplitude of the sinusoidal wave varies like $E^{-1/4}$ at infinity, where E is the kinetic energy of the free ground state rare gas and halogen atoms.

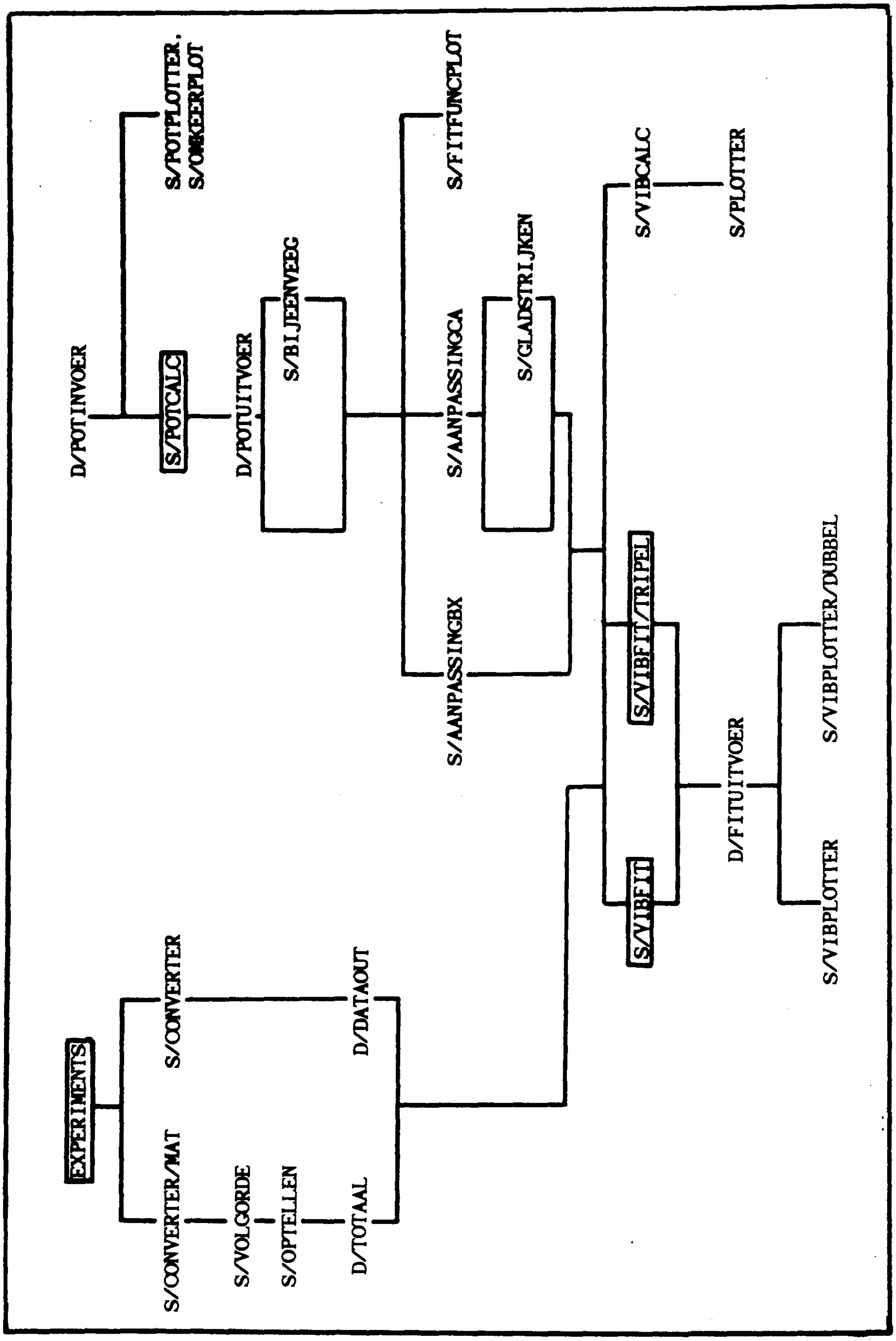

Figure 1: The relation between the available programs for the simulation of bound-free rare gas halide emission spectra.

As input program **S/POTCALC** requires variables giving the potential curves of the upper and lower states and the transition moments, and variables which determine the precision and the size of the calculation which is to be done. Drawing on the comparison between excited state rare gas atoms $(np)^5(n+1)s$ and ground state alkali atoms $(np)^6(n+1)s$, the potential curves are represented by a Rittner-type potential[4], which is given as

$$V(R) = U_1 + U_2 + U_3 \exp(-U_4 R) + U_5 \exp(-U_6 R) - U_7/R - U_8/R^4 - U_9/R^6 \qquad (1)$$

For an explanation of the meaning of the terms in this expression the reader is reffered to Ref. 1.

In the present version of **S/POTCALC** the electronic transition moment is represented as

$$TM(R) = \exp(-\gamma(R-R_e))/ \left(C(1) + C(2)R + C(3)R^2 + C(4)R^3 + C(5)R^4 \right) \qquad (2)$$

However, there are also other representations in use.

All input is usually stored in a file **D/POTINVOER**. An example of a listing of this file along with an explanation of the meaning of all the variables is given below.

FILE **D/POTINVOER**

100	1	If this parameter equals 1, a printer backupfile is made with information about the current program-run.

200	1	If this parameter equals 1, detailed information is written on the before-mentioned printer backupfile about the determination of the bound upper states.
300	2.0,8.0	RMIN, RMAX. These variables determine the interval where wave-functions and contributions to the overlap-integrals have to be calculated.
400	1000	N. The amount of integration-intervals between RMIN and RMAX.
500	0.5,50	EPS, ITMAX. Stopcriterium in cm^{-1} and maximum amount of iterations in the calculation of bound wave-functions.
600	150,1	VMU, STEP. Vibrational quantum number up to which the calculation has to be done and integer interval between vibrational quantum numbers where the calculation has to be done.
700	1500,10	NAC, STEP2. Amount of lower states to be calculated and energy interval in cm^{-1} between consecutive free lower states.
800	130,230	LMIN, LMAX. These variables determine the wavelength interval in nm where the spectra have to be calculated.
900	.2	STEP3. The intervalsize in nm for the calculated spectra.
1000	UPPARR(1)...UPPARR(9)	Parameters to express the upper state potential (see Eq. 1)
1100	LOWARR(1)...LOWARR(9)	Parameters to express the lower state potential (see Eq. 1)
1200	0	If this parameter equals 1, a flat electronic

transition moment is used in the calculation.

1300	GAMMA, C(1)...C(5)	Parameters to express the electronic transition moment. (see Eq. 2)
1400	3	Integer to choose the rare gas atom: Ne=1, Ar=2, Kr=3, Xe=4.
1500	3	Integer to choose the halide atom: F=1, Cl=2, Br=3, I=4
1600	55000	This parameter can be used to choose the vibrational energy in case of a single vibrational level-calculation (VMU=1).

As output program **S/POTCALC** writes the following items to a file **D/POTUITVOER**:

- LMIN, LMAX, STEP3
- VIB, STEP (VIB is equal to the amount of vibrational states for which the calculation has actually been done).
- UMINUP (UMINUP is equal to the minimum value of the upper state potential).
- EU(0:179) (EU is an array in which the vibrational energy-values which have been determined are stored).
- SPEK(0:179, 0:1019) (SPEK is a matrix in which the emission-spectra of the individual vibrational levels are stored).

Calculations with program **S/POTCALC** tend to be very expensive with respect to calculation time. The amount of time required for a calculation is determined mainly by the choice of the parameters N, determining the amount of intervals contributing to the overlap-integral, VIB, the amount of vibrational levels for which the calculation is carried out, and NAC, the amount of free lower states used. In our calculations we adopted the following values:

N=1000

VIB=180

NAC=1300-1500

With this input one complete calculation requires about 3000 secs of processing time on the Burroughs-79 (6000 PT-units).

Program **S/BIJEENVEEG**

In a number of applications it is undesirable to treat all the spectra of individual vibrational levels separately, but rather one wants to group the vibrational states into regular entities. In this case a lot of I/O time is saved if the operation of collecting the vibrational states is done immediately after the calculations in program **S/POTCALC**. The present program transforms a datafile containing calculations done for given values of VIB and STEP, into a new datafile with newly defined values VIB2 and STEP2. Spectra of individual vibrational states are summed to give a new spectrum, which is assigned a vibrational energy which is the average of the vibrational energies of all states included in the summation.

Programs **S/AANPASSINGBX** and **S/AANPASSINGCA**

It is desirable to be able to change the relative shape of the potential curves and the shape of the electronic transition moment without having to resort to a completely new, expensive calculation through program **S/POTCALC**. Using simple semi-classical arguments this is indeed possible[5,6].

Following the classical Franck-Condon principle the electronic transition from an upper state to a lower state happens so fast that both the position and the velocities of the particles involved remain unchanged in the transition. This means, in quantum-mechanical terms, that contributions to the overlap-integral will arise mainly from areas where there is a stationary phase-difference between the upper and lower state wave-functions[7]. Thus the position along the internuclear axis and the wavelength of the emitted photon are strongly related. This implies that a change of the electronic transition moment can be directly related to an intensity change at a corresponding wavelength. And in the case of a change of the relative shape of the potential curves the intensity at a certain wavelength which was calculated for the old set of potential curves can directly be related to the intensity at a corresponding wavelength for the new set of potential curves.

A complication can occur, if there is the possibility of more than one point of stationary phase along the internuclear axis and thus contributions at one wavelength from more than one position along the internuclear axis (see Figure 2), as is the case for rare gas halide B(1/2)-X(1/2) transitions. In this case a classical expression is used to differentiate between the relative probabilities for the emission at a certain wavelength to originate from either of the point of stationary phase as

$$\rho_i(\nu) \sim 1/\left[\sqrt{2m(E_v - V_u(r)}\ \frac{d}{dR}(V_u(R) - V_l(R)) \right]_{R=R_{\nu_i}} \qquad (3)$$

According to this expression the relative emission probability $\rho_i(\nu)$

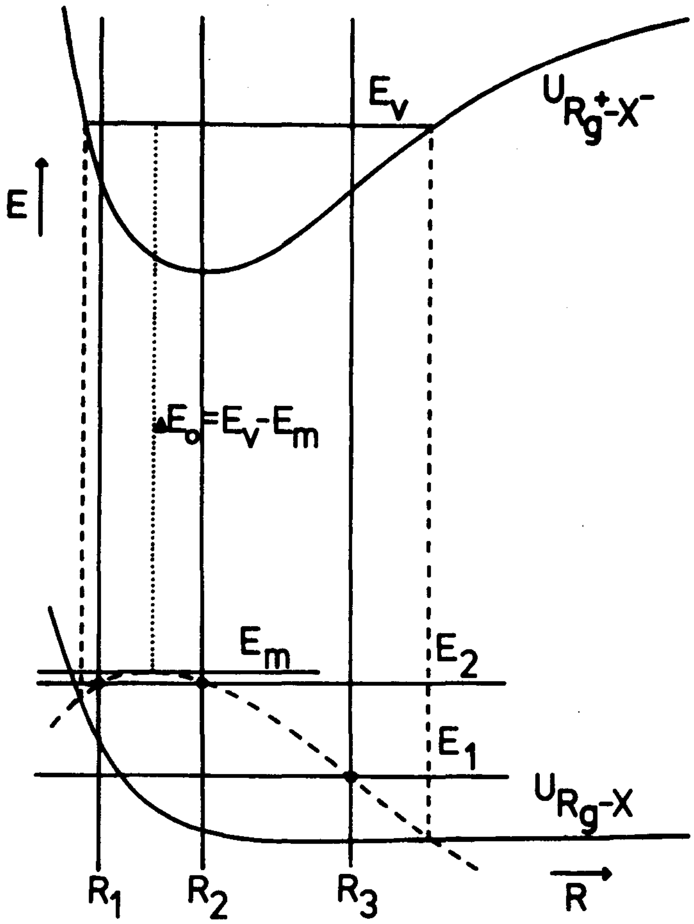

Figure 2: Schematic view of the transition from a vibrational state $|v\rangle$ of a bound Rg^+-X^- ion-pair state to a free Rg-X state. The dashed line is Mulliken's difference potential[10] given as $U_M = U_{Rg-X} - U_{Rg^+-X^-} + E_v$, which gives the location of the points of stationary phase along the internuclear axis. At these points the kinetic energy is conserved during a vertical transition. From this figure it follows that at a free lower state energy E_1 (and a corresponding wavelength $hc/(E_v-E_1)$) there is only one point of stationary phase along the internuclear axis, whereas for a free lower state energy E_2 (and a corresponding wavelength $hc/(E_v-E_2)$) there are two such points. The long wavelength limit of the emission spectrum is determined by the emission to the maximum of Mulliken's difference potential ($\lambda_{max} = hc/(E_v-E_M)$.

is inversely proportional to the velocity of the particles at an internuclear distance R_{υ_i} (and thus proportional to the time-interval spent in the neighbourhoud of R_{υ_i}) and further, inversely proportional to the difference between the gradients of the upper and lower state potential at R_{υ_i}, which means proportional to the width of the interval of stationary phase around R_{υ_i}.

Programs **S/AANPASSINGBX** and **S/AANPASSINGCA** have been written to satisfy needs as they arose in the simulations carried out sofar. Therefore, whereas **S/AANPASSINGCA** has the option of changing both the relative shape of the potential curves and the shape of the electronic transition moment, **S/AANPASSINGBX** has sofar been restricted to the latter. The option of changing the relative shapes of the potential curves can easily be incorporated into the program in the future, when necessary.

Programs **S/AANPASSINGBX** and **S/AANPASSINGCA** require as input a file **D/POTINVOER**, which contains among other things the parameters of the new electronic transition moment and, inside the program, the parameters of the old electronic transition moment. With this information a specified file **D/POTUITVOER** can be transformed into a new D-file.

Program S/GLADSTRIJKEN

With this program calculated C(3/2)-A(3/2) emission spectra from individual vibrational states can be given an envelope which goes down exponentially towards shorter wavelengths. This is achieved by fitting the calculated secondary maxima of the emission spectrum to an exponential function.

Program **S/CONVERTER**

Within the Atomic Collisions and Spectroscopy Group there are standard procedures to transport experimental data to the Burroughs 7900 central EUT-computer, where historically these data are analysed with programs written in Algol. Program **S/CONVERTER** is a small Algol-program which converts a transported data-file to a file **D/DATAOUT**, which is suitable for use in Fortran-programs such as **S/VIBFIT** and **S/VIBFIT/TRIPEL**.

Programs **S/CONVERTER/MAT**, **S/VOLGORDE** and **S/OPTELLEN**

Program **S/CONVERTER/MAT** is very similar to program **S/CONVERTER**. However, where **S/CONVERTER** involves the transportation of merely one data-file, **S/CONVERTER/MAT** is used when a measurement has been repeated several times, as is often done during an experiment. In program **S/CONVERTER/MAT** all the measurements at a certain wavelength (from different scans of the spectrum) are collected and stored together on a file **D/DATAOUT**. The program **S/VOLGORDE** puts all measurements at a certain wavelength in order of magnitude and stores them on a file **D/DATAOUT/2**. Finally program **S/OPTELLEN** generates an experimental emission spectrum, by adding up the individual measurements at each wavelength, with the option of leaving out several values (highest and/or lowest) at each wavelength, to be rid of noise peaks and measurements where the beam source did not function as desired. This final spectrum is stored on a file **D/TOTAAL**.

Programs S/VIBFIT and S/VIBFIT/TRIPEL

The programs **S/VIBFIT** and **S/VIBFIT/TRIPEL** are two programs which are used to derive vibrational state distributions from experimental emission spectra. The only difference between the two programs is the fact that **S/VIBFIT/TRIPEL** offers the possibility to determine the vibrational state distribution simultaneously in several different electronic excimer states, whereas **S/VIBFIT** is applicable in the case of only one electronic excimer state. Thus **S/VIBFIT** can be used to get a rough idea of the vibrational state distributions, whereas **S/VIBFIT/TRIPEL** offers the opportunity to eliminate errors due to overlapping emissions from more than one electronic transition. In the present form of **S/VIBFIT/TRIPEL** an experimental spectrum can be fitted to three different electronic transitions, with the restriction that these origininate from two different upper states. This obviously arises from the fact that this program has been written in order to do simultaneous simulations of B(1/2)-X(1/2), C(3/2)-A(3/2) and B(1/2)-A(1/2) continua. It goes without saying that in the near future this program can and should be further extended in order to include yet more bound-free excimer transitions, and a number of atomic and molecular emissions due to competing inelastic reaction channels (cf. the fairly large Br_2^* emission which is encountered in the reactions of Kr $(^3P_{0,2})$+Br_2 and Xe $(^3P_{0,2})$+Br_2).

The vibrational state distributions are determined through a linear least squares analysis. This least squares method is based on the following matrix-equation[8,9]

$$\overline{D} = \overline{S}.\overline{P} \qquad (4)$$

where $\overline{D}(0:NMEETP-1)$ is an array which contains the experimental spectrum, $\overline{P}(0:NPAR-1)$ is an array which contains the vibrational state distribution to be determined and $\bar{\bar{S}}(0:NPAR-1, 0:NMEETP-1)$ is a matrix in which the spectra of the individual vibrational levels are stored, which serve as basis functions in the least squares analysis.

The least squares solution $\overline{P}(0:NPAR-1)$ is determined by the condition that

$$R = \sum_{i=0}^{NMEETP-1} [\, \overline{D}(i) - \sum_{j=0}^{NPAR-1} \bar{\bar{S}}(i,j)\, \overline{P}(j) \,]^2 \tag{5}$$

be minimal. This condition requires that

$$\frac{\delta R}{\delta \overline{P}(k)} = -2 \sum_{i=0}^{NMEETP-1} \bar{\bar{S}}(i,k) \, [\, \overline{D}(i) - \sum_{j=0}^{NPAR-1} \bar{\bar{S}}(i,j)\, \overline{P}(j) \,] = 0 \tag{6}$$

These are the Gaussian normal equations, which can be solved using standard matrix algebra.

When using the programs **S/VIBFIT** and **S/VIBFIT/TRIPEL** it is important to limit to some extent the number of basis functions and thus the number of fit-parameters. In our experiments vibrational levels up to v=180 may be occupied, but because of experimental errors in the data and uncertainties in the validity of the calculated spectra from single vibrational levels, a fit with 180 fit-parameters and basis functions will yield a completely unphysical vibrational distribution. Rather, a number of vibrational levels should be taken together in the calculation (within **S/VIBFIT** and **S/VIBFIT/TRIPEL** or beforehand, using **S/BIJEENVEEG**), the average of their spectra being used as one basis function.

In Figure 3 the relation between the uncertainties in the experimental data and the amount of vibrational levels which are taken together is shown. In the calculations leading to this figure experimental data were generated by mapping a flat vibrational distribution on a given set of 150 spectra from single vibrational levels from the XeBr B(1/2)-X(1/2) transition. These exact data, normalized to a maximum value of 10000 'counts', were perturbed in the

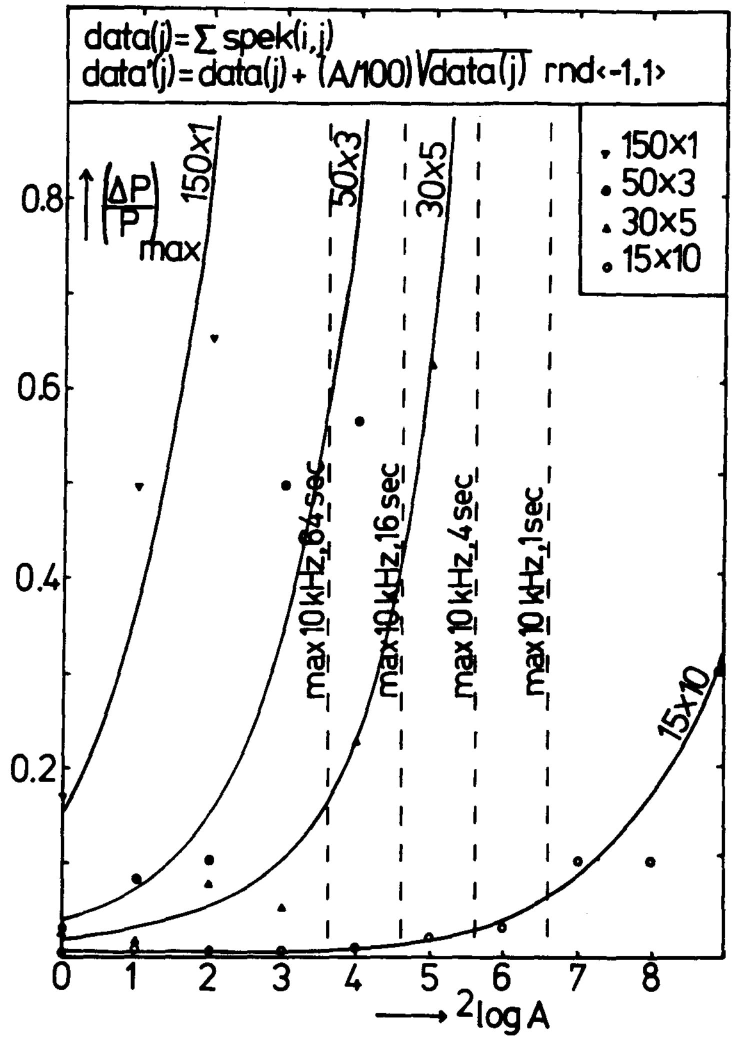

Figure 3: The relationship between the uncertainties in the experimental data and the amount of vibrational levels which are taken together in a least squares fit of an emission spectrum. See text for detailed explanation.

following way,

$$\overline{D}(i)' = \overline{D}(i) \left[1 + A \sqrt{D(i)}\ \mathrm{RND}/100 \right] \qquad (7)$$

where RND is a randum number between -1 and 1. Thus, for example, A=100 would correspond to an uncertainty of approximately one standard deviation for an experiment where the maximum amount of counts at a certain wavelength is equal to 10000. Figure 3 shows the maximum relative deviation from a flat vibrational state distribution as a function of A and the amount of vibrational levels taken together in the calculation. Thus, since we can calculate the value of the parameter A from our experimental conditions, we have a recipy to determine how many vibrational levels should produce one basis function in the least squares fit. It should be stressed that the use of this recipy is conditional on the correctness of the available potential curves and electronic transition moment, which often turns out to be a major problem.

Finally, the structure of the programs **S/VIBFIT** and **S/VIBFIT/TRIPEL** is such that when a data-file **D/DATAOUT** and one (or more) data-files **D/POTUITVOER** are available, the program proceeds through an interactive input of the amount of vibrational states to be used and the amount of vibrational levels to be taken together in the calculation.

Both programs create a previewfile which contains a comparison of the experimental and the simulated spectrum and a file **D/FITUITVOER** where the vibrational state distributions which were determined are stored.

Programs S/VIBPLOTTER and S/VIBPLOTTER/DUBBEL

These programs give a graphical representation of the vibrational state distributions determined in the programs S/VIBFIT and S/VIBFIT/TRIPEL, using the information which was stored on file D/FITUITVOER. As an example, Figure 4 shows some typical output of S/VIBFIT/DUBBEL.

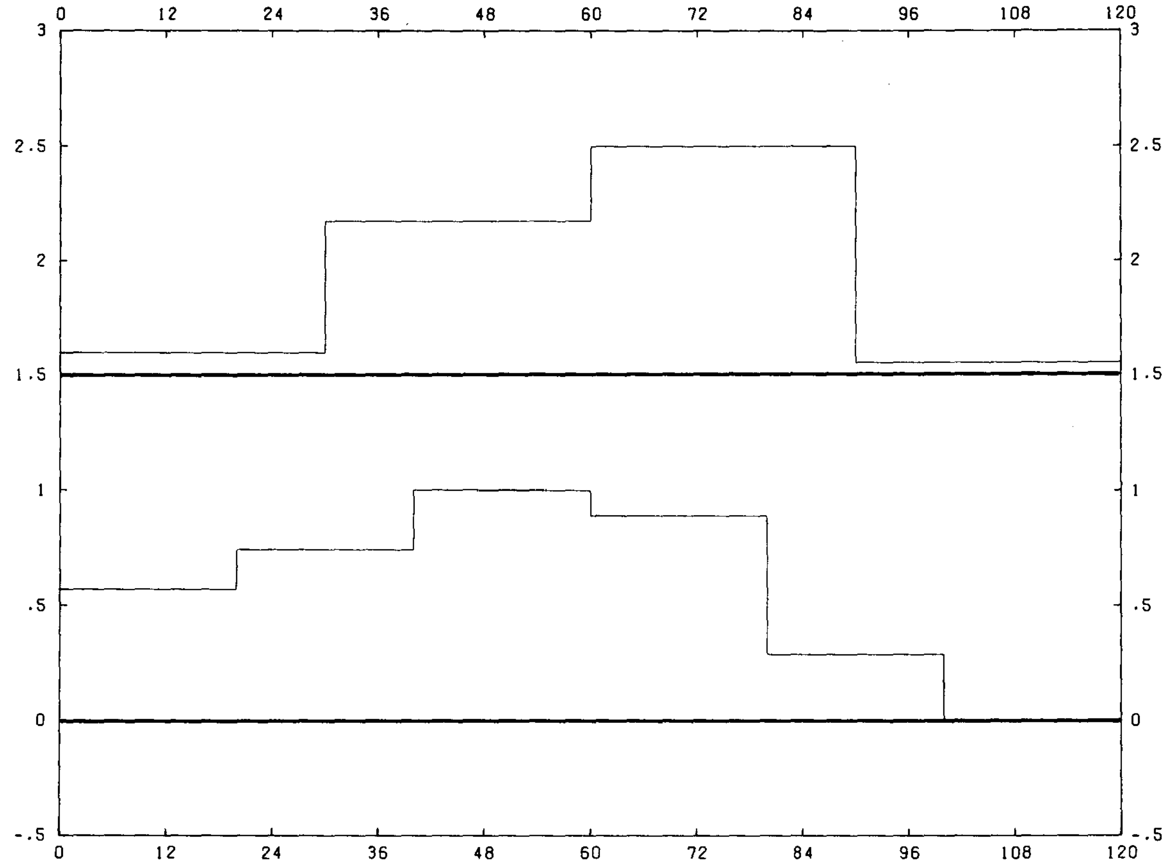

Figure 4: Some typical output of program S/VIBPLOTTER/DUBBEL: the XeBr B(1/2) state (lower curve) and C(3/2) state (upper curve) vibrational distributions determined from the emission spectrum from the reaction of Xe^* + CF_2Br_2. For detailed information on the potential curves and transition moment used see ref. 6.

Programs **S/VIBCALC** and **S/PLOTTER**

Program **S/VIBCALC** can be used to generate a total emission-spectrum starting from given spectra for the individual vibrational levels, calculated in program **S/POTCALC**. The program offers the possibility of applying a number of different vibrational state distributions, including linear surprisal distributions, Boltzmann-distributions and a flat distribution.

As input **S/VIBCALC** requires a file **D/POTUITVOER** and a file **D/VIBINVOER**. A typical listing of **D/VIBINVOER** along with an explanation of the variables is given below.

FILE **D/VIBINVOER**

100	1	If this parameter equals 1, a printer backupfile is made with information about the current program-run.
200	1	KEUS. Integer to choose the type of vibrational state distribution: 1) Linear surprisal distribution. 2) Flat distribution 3) Linear distribution: P(I)=1+C1*EU(I)/EMAX 4) Boltzmann distribution
300	-8	Depending on the value of KEUS this variable represents the linear surprisal parameter λ_v, a dummy, the parameter C1 or the Boltzmann-temperature.
400	45.8	Bonding energy of the halide atom in the target gas molecule (kcal/mol).
500	8.81	Rare gas metastable state energy (eV).

Program **S/VIBCALC** generates a spectrum which is stored on a file **D/VIBUITVOER**. A graphical representation of this spectrum is directly available with the plotprogram **S/PLOTTER**. This program is also often used to plot experimental spectra, stored on **D/DATAOUT** or **D/TOTAAL**. As an example Figure 5 shows the spectrum of Xe^* + CBr_4 plotted with **S/PLOTTER**.

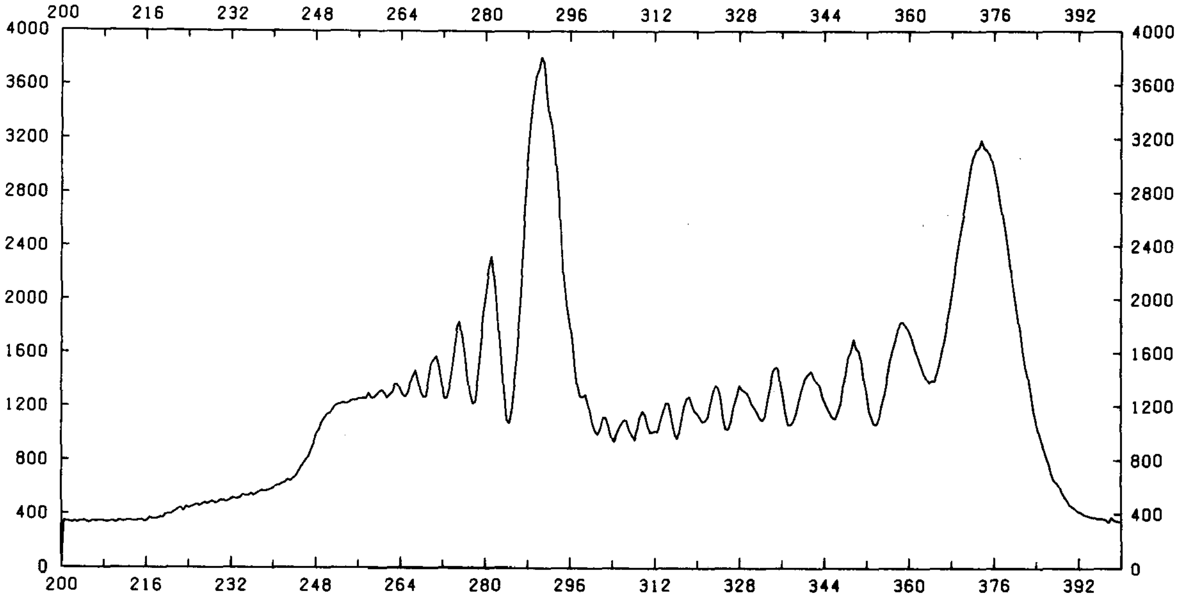

Figure 5: The XeBr emission spectrum from the reaction of Xe^* + CBr_4, plotted using program S/PLOTTER.

Program S/POTPLOTTER

This program, using a file **D/POTINVOER**, gives a graphical representation of a set of potential curves, including Mulliken's difference potential[10] at a specified energy. As an example Figure 6 shows the potential curves of the XeBr B(1/2)-X(1/2) transition, with Mulliken's difference potential for E=55000 cm^{-1}.

Figure 6: The potential curves for the XeBr B(1/2)-X(1/2) transition[5], with Mulliken's difference potential[10] for E=55000 cm^{-1}, plotted using program S/POTPLOTTER.

Program **S/OMKEERPLOT**

This program, using a file **D/POTINVOER** can be used to give a graphical representation of the energy-dependence of the wavelengths of the emission at the classical turning points for a number of specified lower state potentials. As an example Figure 7 shows the emission from the turning points in the XeBr C(3/2)-A(3/2) transition, using the ab-initio C(3/2) state potential curve and a number of trial A(3/2) state potential curves.

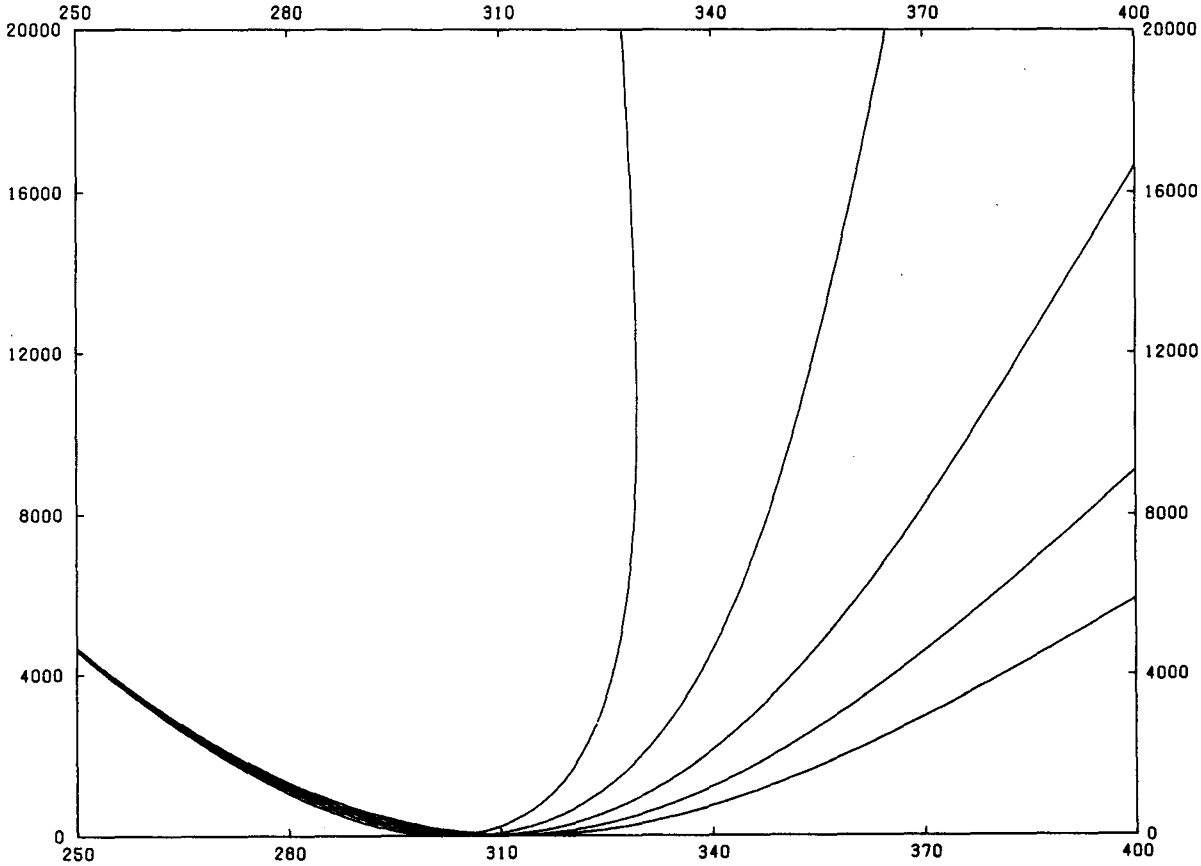

Figure 7: Emission from the turning points in the XeBr C(3/2)-A(3/2) transition, using the ab-initio C(3/2) state potential curve given in ref. 3, and an A(3/2) state potential represented as $V_{A(3/2)}(r) = A \exp(- 2.9464 R) + 600$ [cm^{-1}], where A takes on the values 6.0E7, 5.5E7, 5.0E7, 4.5E7 and 4.0E7, respectively. In the final A(3/2) state potential which was determined from computer simulations of several XeBr emission spectra a value of A of 4.8536E7 was adopted (see ref. 6).

Program S/FITFUNCPLOT

This program can be used to get a graphical representation of the fit-functions used in programs **S/VIBFIT** and **S/VIBFIT/TRIPEL**. As an example Figure 8 shows the fit-functions used in the simulations of XeBr C(3/2)-A(3/2) continua.

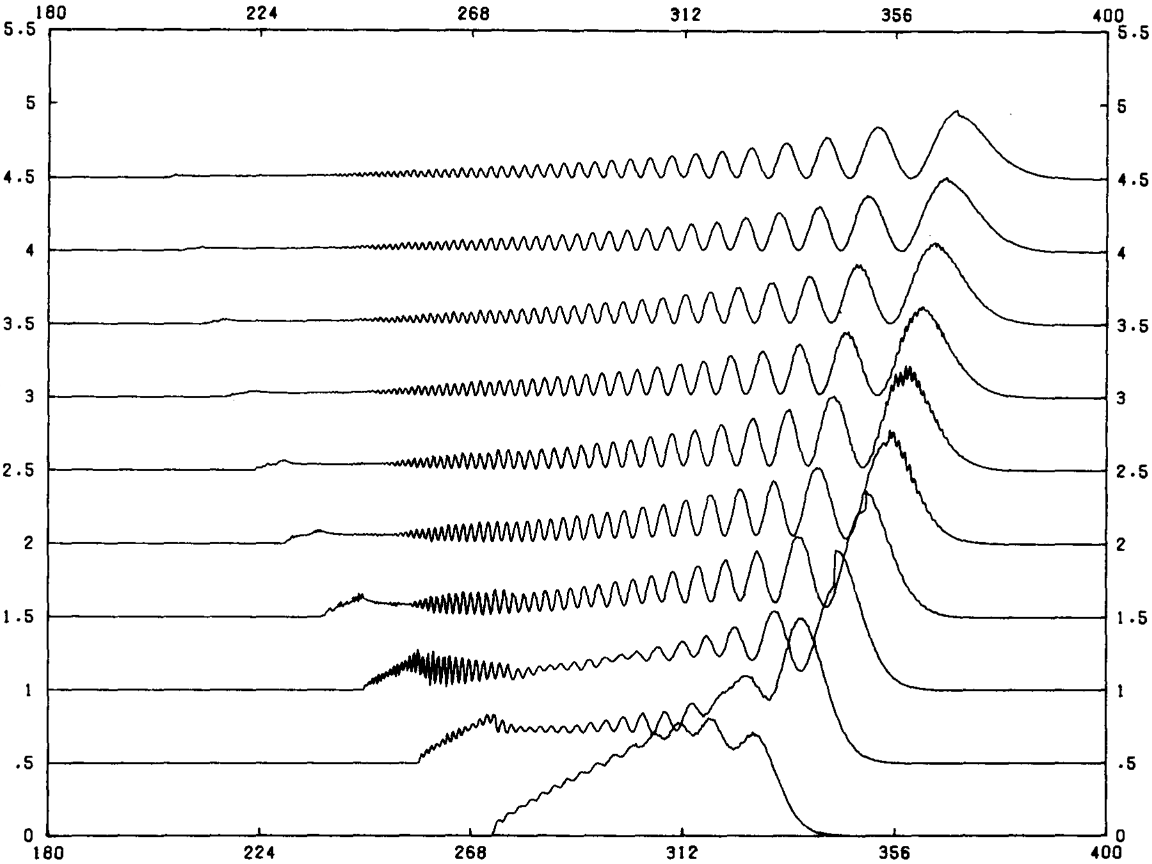

Figure 8: Fit-functions used in the simulations of XeBr C(3/2)-A(3/2) continua, plotted using program S/FITFUNCPLOT. See ref. 6 for the parameters used to represent the potential curves and the electronic transition moment.

REFERENCES

1) M.J.J. Vrakking, K. Tabayashi and K. Shobatake, to be published.

2) J.W. Cooley, Math. Computation XV, 363 (1961)

3) J.K. Cashion, J. Chem. Phys. 39, 1872 (1963)

4) E.S. Rittner, J. Chem. Phys. 19, 1030 (1951)

5) K. Tamagake, J.H. Koltz and D.W. Setser, J. Chem. Phys. 71, 1264 (1979)

6) M.J.J. Vrakking, EUT Internal Report VDF/NO 87-17, see Chapter 2.

7) J. Tellinghuisen, Adv. Chem. Phys. LX, 299 (1985)

8) K. Johnson, A. Kvaran and J.P. Simons, Mol. Phys. 50, 981 (1983)

9) K. Johnson, J.P. Simons, P.A. Smith and C. Washington, Mol. Phys. 57, 255 (1986)

10) R.S. Mulliken, J. Chem. Phys. 55, 309 (1977)

```
MINIB2/S/GEBONDENVRIJ/POTCALC
DATE & TIME PRINTED: MONDAY, JULY 20, 1987 @ 10:46:47.


C     MINIB2/S/GEBONDENVRIJ/POTCALC
C
C     This program is part of a series of programs to simulate
C     bound-free rare gas halide emission spectra.
C     In this program the emission spectra of individual vibrational
C     upper states are determined by calculating the overlap
C     integrals of the radial parts of the bound upper state wave-
C     functions and the free lower state wave-functions.
C     The input variables, describing the potential curves
C     and the electronic transition moment, and the precision
C     and the size of the calculation, are usually stored on
C     a specified FILE 4. The spectra are stored on a specified
C     FILE 5.
C
C*********************************************************************
C
C     DECLARATIONS
C     ************
C
FILE  1(KIND=REMOTE)                    % WRITE TO TERMINAL
FILE  2(KIND=REMOTE)                    % READ FROM TERMINAL
FILE  3(KIND=PRINTER)
FILE  4(KIND=DISK,FILETYPE=7,TITLE="MINIB2/D/GEBONDENVRIJ/POTINVOER")
FILE  5(KIND=DISK,MAXRECSIZE=30,BLOCKSIZE=480,AREASIZE=30,
     *   TITLE="MINIB2/D/GEBONDENVRIJ/POTUITVOER",
     *   PROTECTION=SAVE,NEWFILE=TRUE)
      REAL*4 RMIN,RMAX,EPS,STEP2,UPPARR,LOWARR,DUMMY,GAMMA,C1,
     *    C2,C3,C4,C5,M1,M2,MURED,H,UMINUP,UMINLO,R,UUP,ULO,REUP,TM,E,
     *    E1,FAC,AMP,Y,EE,FCF,STEP3,REALJB,HULP,SPEK,NU,OMEUUP,PSU,EU
      INTEGER*4 INTACT,TERM,N,ITMAX,VMU,STEP,NAC,SQUEST,SEDEL,SHALO,
     *    I,J,K,TERM2,VIB,JMIN,JMAX,INTJB,LMIN,LMAX,KMAX,TRANS
      DIMENSION UPPARR(9),LOWARR(9),R(0:1001),UUP(0:1001),ULO(0:1001),
     *    TM(0:1001),PSU(1:180,0:1001),EU(180),FCF(1:180,1:1500),
     *    Y(0:1001),SPEK(1:180,0:1019)
C
C     INPUT VARIABLES
C     ***************
C
      SQUEST=0
      INTACT=4
      WRITE(1,1)
      READ(2,/) SQUEST
    1 FORMAT(´ INVOER INTERACTIEF VANAF TERMINAL (JA=1)?´)
C 1   FORMAT(´ INPUT INTERACTIVE FROM TERMINAL (YES=1)?´)
      IF(SQUEST.EQ.1) INTACT=2
C
      SQUEST=0
      TERM=0
      IF(INTACT.EQ.2) WRITE(1,2)
      READ(INTACT,/) SQUEST
    2 FORMAT(´ AANMAAK PRINTERFILE MET PROGRAMMAGEGEVENS (JA=1)?´)
C 2   FORMAT(´ CREATE A PRINTERFILE WITH PROGRAM INFO (YES=1)?´)
      IF(SQUEST.EQ.1) TERM=1
      SQUEST=0
      TERM2=0
      IF(TERM.EQ.0) READ(INTACT,/) DUMMY
      IF(TERM.EQ.0) GOTO 22
      IF(INTACT.EQ.2) WRITE(1,21)
```

```
   21 FORMAT(' UITVOERIGE GEGEVENS OVER GEBONDEN TOESTANDEN (JA=1)?')
C  21 FORMAT(' EXTENSIVE INFO ON BOUND UPPER STATES (YES=1)?')
      READ(INTACT,/) SQUEST
      IF(SQUEST.EQ.1) TERM2=1
   22 CONTINUE
C
      IF(INTACT.EQ.2) WRITE(1,3)
      READ(INTACT,/) RMIN,RMAX
    3 FORMAT(' INVOER RMIN, RMAX IN ANGSTROEM: ')
C   3 FORMAT(' INPUT OF RMIN, RMAX IN ANGSTROEM: ')
      IF(INTACT.EQ.2) WRITE(1,4)
      READ(INTACT,/) N
    4 FORMAT(' INVOER AANTAL INTEGRATIE STAPPEN N: ')
C   4 FORMAT(' INPUT AMOUNT OF INTEGRATION STEPS N: ')
      IF(INTACT.EQ.2) WRITE(1,5)
      READ(INTACT,/) EPS,ITMAX
    5 FORMAT(' INVOER BEEINDIGINGSCRITERIA EPS, ITMAX: ')
C   5 FORMAT(' INPUT STOPCRITERIA EPS, ITMAX: ')
      IF(INTACT.EQ.2) WRITE(1,6)
      READ(INTACT,/) VMU,STEP
    6 FORMAT(' INVOER AANTAL BOVENNIVEAUS EN STAPGROOTTE: ')
C   6 FORMAT(' INPUT AMOUNT OF UPPER STATES AND INTERVALSIZE: ')
      IF(INTACT.EQ.2) WRITE(1,7)
      READ(INTACT,/) NAC,STEP2
    7 FORMAT(' INVOER AANTAL BENEDENNIVEAUS EN STAPGROOTTE: ')
C   7 FORMAT(' INPUT AMOUNT OF LOWER STATES AND INTERVALSIZE: ')
      IF(INTACT.EQ.2) WRITE(1,71)
      READ(INTACT,/) LMIN,LMAX
   71 FORMAT(' INVOER GOLFLENGTE INTERVALGRENZEN LMIN,LMAX: ')
C  71 FORMAT(' INPUT BOUNDARIES WAVELENGTH INTERVAL LMIN, LMAX: ')
      IF(INTACT.EQ.2) WRITE(1,72)
      READ(INTACT,/) STEP3
   72 FORMAT(' INVOER STAPGROOTTE IN GOLFLENGTEINTERVAL: ')
C  72 FORMAT(' INPUT STEPSIZE IN WAVELENGTH INTERVAL: ')
      IF(INTACT.EQ.2) WRITE(1,8)
      READ(INTACT,/) UPPARR(1),
     *    UPPARR(2),UPPARR(3),UPPARR(4),UPPARR(5),UPPARR(6),
     *    UPPARR(7),UPPARR(8),UPPARR(9)
    8 FORMAT(' INVOER POTENTIAALGEGEVENS UPPARR(1)...UPPARR(9): ')
C   8 FORMAT(' INPUT POTENTIAL PARAMETERS UPPARR(1)...UPPARR(9): ')
      IF(INTACT.EQ.2) WRITE(1,9)
      READ(INTACT,/) LOWARR(1),
     *    LOWARR(2),LOWARR(3),LOWARR(4),LOWARR(5),LOWARR(6),
     *    LOWARR(7),LOWARR(8),LOWARR(9)
    9 FORMAT(' INVOER POTENTIAALGEGEVENS LOWARR(1)...LOWARR(9): ')
C   9 FORMAT(' INPUT POTENTIAL PARAMETERS LOWARR(1)...LOWARR(9): ')
C
      SQUEST=0
      TRANS=0
      IF(INTACT.EQ.2) WRITE(1,10)
      READ(INTACT,/) SQUEST
   10 FORMAT(' CONSTANT OVERGANGSMOMENT (JA=1)?')
C  10 FORMAT(' FLAT TRANSITION MOMENT (YES=1)?')
      IF(SQUEST.EQ.0) TRANS=1
      IF(TRANS.EQ.0.AND.INTACT.EQ.4) READ(INTACT,/) DUMMY,DUMMY,
     *    DUMMY,DUMMY
      IF(TRANS.EQ.1.AND.INTACT.EQ.2) WRITE(1,11)
   11 FORMAT(' INVOER OVERGANGSMOMENT GAMMA,C1,C2,C3,C4,C5: ')
C  11 FORMAT(' INPUT TRANSITION MOMENT GAMMA,C1,C2,C3,C4,C5: ')
      IF(TRANS.EQ.1) READ(INTACT,/) GAMMA,C1,C2,C3,C4,C5
C
      IF(INTACT.EQ.2) WRITE(1,12)
```

```
      READ(INTACT,/) SEDEL
  12  FORMAT(' INVOER EDELGAS NE=1, AR=2, KR=3, XE=4: ')
C 12  FORMAT(' INPUT RARE GAS NE=1, AR=2, KR=3, XE=4: ')
      IF(INTACT.EQ.2) WRITE(1,13)
      READ(INTACT,/) SHALO
  13  FORMAT(' INVOER HALOGEEN F=1, CL=2, BR=3, I=4: ')
C 13  FORMAT(' INPUT HALOGEN F=1, CL=2, BR=3, I=4: ')
C
      IF(SEDEL.EQ.1) M1=20.183
      IF(SEDEL.EQ.2) M1=39.948
      IF(SEDEL.EQ.3) M1=83.80
      IF(SEDEL.EQ.4) M1=131.30
      IF(SHALO.EQ.1) M2=18.998
      IF(SHALO.EQ.2) M2=35.453
      IF(SHALO.EQ.3) M2=79.909
      IF(SHALO.EQ.4) M2=126.9044
      MURED=M1*M2/(M1+M2)
C
      IF(TERM.EQ.1) CALL VARCOM(RMIN,RMAX,N,VMU,STEP,NAC,STEP2,
     *    UPPARR,LOWARR,GAMMA,C1,C2,C3,C4,C5,MURED)
C
C     CALCULATION OF POTENTIAL CURVES AND TRANSITION MOMENTS
C     *****************************************************
C
      H=(RMAX-RMIN)/(N+1)
      UMINUP=10**10
      UMINLO=10**10
C
      DO 30 I=0,N+1,1
      R(I)=RMIN+I*H
C     R(I)=R(I)*1.09       Note: for XeBr B-X(1/2) calculation!
      UUP(I)=UPPARR(1)+UPPARR(2)
     *   +UPPARR(3)*EXP(-UPPARR(4)*R(I))
     *   +UPPARR(5)*EXP(-UPPARR(6)*R(I))
     *   -UPPARR(7)/R(I)
     *   -UPPARR(8)/(R(I)**4)
     *   -UPPARR(9)/(R(I)**6)
      ULO(I)=LOWARR(1)+LOWARR(2)
     *   +LOWARR(3)*EXP(-LOWARR(4)*R(I))
     *   +LOWARR(5)*EXP(-LOWARR(6)*R(I))
     *   -LOWARR(7)/R(I)
     *   -LOWARR(8)/(R(I)**4)
     *   -LOWARR(9)/(R(I)**6)
      IF(UUP(I).GT.UMINUP) GOTO 20
      UMINUP=UUP(I)
      REUP=R(I)
  20  IF(ULO(I).GT.UMINLO) GOTO 25
      UMINLO=ULO(I)
  25  CONTINUE
C     R(I)=R(I)/1.09       Note: for XeBr B-X(1/2) calculation!
  30  CONTINUE
C
      DO 40 I=0,N+1,1
      TM(I)=1.
      IF(TRANS.EQ.1) TM(I)=EXP(GAMMA*(R(I)-REUP))/
C    *    ((R(I)-C1)**2+C2) + C3
     *    (C1+C2*R(I)+C3*(R(I)**2)+C4*(R(I)**3)+C5*(R(I)**4))
  40  CONTINUE
C
C
C     CALCULATION OF BOUND UPPER STATES
C     *********************************
```

```
C
      OMEUUP=150.
      E=UMINUP+OMEUUP/2
      IF(VMU.NE.1) GOTO 50
      IF(INTACT.EQ.2) WRITE(3,14)
   14 FORMAT(' TRIAL-ENERGIEWAARDE VOOR DE BOVENTOESTAND: ')
C  14 FORMAT(' TRIAL ENERGY VALUE FOR UPPER STATE')
C
C     If VMU equals 1 the spectrum of only one vibrational level
C     will be calculated. In this case the calculation does not
C     start from the well of the upper state potential curve, but
C     a vibrational energy can be specified by the user.
C
   45 READ(INTACT,/)E1
      IF(E1.LT.E) WRITE(3,15)
   15 FORMAT(' TRIAL-ENERGIEWAARDE LIGT BENEDEN POTENTIAALMINIMUM',/,
     *     ' VOER EEN HOGERE WAARDE IN!')
C  15 FORMAT(' TRIAL ENERGY VALUE LIES BELOW THE MINIMUM OF THE',/,
C    *     ' UPPER STATE POTENTIAL CURVE',/,
C    *     ' PLEASE ENTER A HIGHER VALUE!')
      IF(E1.LT.E) GOTO 45
      E=E1
   50 CONTINUE
C
      IF(TERM.EQ.1) WRITE(3,16) UMINUP,REUP
   16 FORMAT(/,' UMINUP= ',E14.7,/,' REUP= ',F9.4,/)
      WRITE(1,/) TIME(2)/60.
      CALL BWAVE(1,DUMMY,VMU,STEP,UUP,R,N,E,OMEUUP,MURED,EPS,
     *     ITMAX,VIB,EU,PSU,TERM2)
C
C     Subroutine BWAVE is the subroutine where the actual calculation
C     of the upper state wave-functions takes place. An explanation of
C     the parameters in the heading of BWAVE is included in the
C     source text of BWAVE.
C
      WRITE(1,/) TIME(2)/60.
      IF(TERM.EQ.1.AND.VMU.NE.1) WRITE(3,17) (EU(2)-EU(1))/STEP
   17 FORMAT(/,' BEREKENDE OMEGA IN BOVENTOESTAND: ',F9.2,/)
C  17 FORMAT(/,' CALCULATED OMEGA IN UPPER STATE: 'F9.2,/)
C
C     CALCULATION OF FREE LOWER STATES AND FRANCK-CONDON FACTORS
C     **********************************************************
C
C     Initialization
C
      FAC=MURED/16.86
      DO 120 I=1,NAC,1
      E=LOWARR(1)+LOWARR(2)+(I-.5)*STEP2
      EE=E**.25*N**.5
C
      JMAX=N
      JMIN=1
      DO 60 J=1,N-5,1
      IF(JMIN.EQ.1.AND.ULO(J+5).LE.E) JMIN=J
   60 CONTINUE
C
      DO 70 J=1,N,1
      Y(J)=0
   70 CONTINUE
C
C     Integration
C
```

```
      Y(JMIN+1)=1.
      DO 80 J=JMIN+2,N,1
      Y(J)=Y(J-1)*(2+FAC*H*H*(ULO(J-1)-E)/
     *   (1-FAC*H*H*(ULO(J-1)-E)/12))-Y(J-2)
   80 CONTINUE
C
C     Determination of wave-function amplitudo ´at infinity´
C
      K=JMAX-1
      AMP=0.
   90 IF(ABS(Y(K-1)).LE.ABS(Y(K)).AND.ABS(Y(K)).GT.ABS(Y(K+1))) AMP=
     *   ABS(Y(K))
      IF(AMP.NE.0) GOTO 95
      K=K-1
      IF(K.GT.1) GOTO 90
      WRITE(1,18)
   18 FORMAT(´ ERROR: PROGRAMMA WORDT GESTAAKT OMDAT HET INTEGRATIE´,
     *   ´INTERVAL TE KLEIN IS VOOR HET BEREKENEN VAN BETROUWBARE´,
     *   ´VRIJ GOLFFUNCTIES. DE WAARDE VAN RMAX DIENT TE WORDEN´,
     *   ´VERHOOGD.´)
C  18 FORMAT(´ ERROR: PROGRAM STOPPED BECAUSE THE INTEGRATION´,
C    *   ´INTERVAL IS TOO SMALL FOR A CALCULATION OF RELIABLE FREE´,
C    *   ´LOWER STATE WAVE-FUNCTIONS. THE VALUE OF RMAX HAS TO´,
C    *   ´BE ENLARGED.´)
      GOTO 999
C
C     Normalization
C
   95 DO 100 J=2,N+1,1
      Y(J)=Y(J)/AMP/EE
  100 CONTINUE
C
C     Calculation of Franck-Condon factors
C
      DO 110 J=1,VIB,1
      FCF(J,I)=0.
      DO 105 K=2,N,1
      FCF(J,I)=FCF(J,I)+PSU(J,K)*TM(K)*Y(K)
  105 CONTINUE
      FCF(J,I)=FCF(J,I)**2
  110 CONTINUE
C
  120 CONTINUE
      WRITE(1,/) TIME(2)/60.
C
C     DETERMINATION OF THE EMISSION SPECTRA OF INDIVIDUAL
C     VIBRATIONAL STATES
C     ***************************************************
C
C
      KMAX=INT((LMAX-LMIN)/STEP3)
      DO 140 I=1,VIB,1
      DO 130 K=0,KMAX,1
      NU=(10**7)/(LMIN+K*STEP3)
      REALJB=.5+(EU(I)-LOWARR(1)-LOWARR(2)-NU)/STEP2
      INTJB=INT(REALJB)
      IF(INTJB.GT.NAC-1) GOTO 130
      IF(INTJB.LE.0) GOTO 130
      HULP=REALJB-INTJB
      SPEK(I,K)=(NU*5.)*((1-HULP)*FCF(I,INTJB)+HULP*FCF(I,INTJB+1))
  130 CONTINUE
  140 CONTINUE
```

```
C
C     Output to a file D/POTUITVOER
C
      WRITE(5,/) LMIN,LMAX,STEP3
      WRITE(5,/) VIB,STEP
      WRITE(5,/) UMINUP
      WRITE(5,/) EU
C
      DO 161 I=1,180
      DO 151 J=1,34
      WRITE(5) (SPEK(I,30*(J-1)+K),K=0,29)
  151 CONTINUE
  161 CONTINUE
C
  999 CONTINUE
      CLOSE(5,DISP=CRUNCH)
C     Crunching of outputfile D/POTUITVOER
C
      END
C
      SUBROUTINE BWAVE(MODE,EMAX,VMU,STEP,U,R,N,E,OMEU,MURED,EPS,ITMAX,
     *    VIBC,EU,PSI,TERM)
C
C     In this subroutine the radial parts of the upper state wave-
C     functions are calculated, using an algorithm developed by
C     Cooley (Math. Computation XV, 363 (1961)).
C     If the parameter MODE equals 1 then vibrational states are
C     determined up to v=VMU, at an interval STEP. If the parameter
C     MODE equals 2, then vibrational states are determined up to
C     an energy EMAX.
C     The array U gives the potential energy at an internuclear
C     distance specified by array R, the parameter E is a trial
C     energy value for the first vibrational state to be determined,
C     and OMEU is an estimate of the OMEGA-value of the potential
C     curve. The parameters N, EPS, ITMAX and MURED have the meaning
C     introduced in the main source program.
C     As output an integer VIBC, representing the amount of vibra-
C     tional states determined, an array EU, containing the energy
C     eigenvalues, and a matrix PSI, containing the radial wave-
C     functions are given.
C     If the parameter TERM is given the value 1, then a printerfile
C     is created with extensive information on the calculation of
C     the radial wave-functions.
C
C     DECLARATIONS
C     ************
C
      REAL*4 FAC,MURED,U,E,H,R,Y,SOM,DE,EPS,EU,OMEU,EMAX,PSI
      INTEGER*4 VIB,VIBC,STEP,IT,JMIN,JMAX,N,M,NO,INO,MODE,VMU,
     *    ITMAX,TERM,J,JINV
      DIMENSION U(0:1001),R(0:1001),Y(0:1001),EU(180),PSI(1:180,0:1001)
C
C     INITIALIZATION
C     **************
C
      IF(TERM.EQ.1) WRITE(3,2)
    2 FORMAT(//,' *** CALCULATIE VAN GEBONDEN GOLFFUNCTIES ***')
C   2 FORMAT(//,' *** CALCULATION OF BOUND WAVE-FUNCTIONS ***')
      FAC=MURED/16.86
      VIB=0
      VIBC=0
      IT=0
```

```
      H=R(2)-R(1)
    5 IF (TERM.EQ.1) WRITE(3,8) E
    8 FORMAT(/,' TRIAL-ENERGIE= ',E14.7,//,
     * '  IT    M        R(M)               E            DE')
C 8   FORMAT(/,' TRIAL ENERGY VALUE= ',E14.7,//,
C    * '  IT    M        R(M)               E            DE')
   25 JMIN=1
      JMAX=N+1
      DO 10 J=1,N-24,1
      IF(JMIN.EQ.1.AND.U(J+25).LE.E) JMIN=J
   10 CONTINUE
      DO 20 JINV=26,N+1,1
      J=N+27-JINV
      IF(JMAX.EQ.N+1.AND.U(J-25).LE.E) JMAX=J
   20 CONTINUE
C
      Y(JMIN)=0
      Y(JMIN+1)=(1-FAC*H*H*(U(JMIN+1)-E))
      Y(JMAX)=(1-FAC*H*H*(U(JMAX)-E))
      Y(JMAX-1)=(1-FAC*H*H*(U(JMAX-1)-E))*EXP(SQRT(FAC)*
     *    (R(JMAX)*SQRT(U(JMAX)-E)-R(JMAX-1)*SQRT(U(JMAX-1)-E)))
C
C     INWARD INTEGRATION
C     ******************
C
      M=0
      Y(JMAX-2)=H*H*Y(JMAX-1)*(1/H/H+FAC*(U(JMAX-1)-E)/
     *    (1-FAC*H*H*(U(JMAX-1)-E)/12))
C
      DO 40 JINV=JMIN+1,JMAX-3,1
      J=JMIN+JMAX-2-JINV
      IF(M.NE.0) GOTO 40
      Y(J)=Y(J+1)*(2+FAC*H*H*(U(J+1)-E)/
     *    (1-FAC*H*H*(U(J+1)-E)/12))-Y(J+2)
      IF(ABS(Y(J)).LE.ABS(Y(J+1))) M=J
   40 CONTINUE
C
      IF(Y(M).EQ.0) M=M+1
      DO 50 JINV=M,JMAX,1
      J=M+JMAX-JINV
      Y(J)=Y(J)/Y(M)
   50 CONTINUE
C
C     OUTWARD INTEGRATION
C     *******************
C
      DO 60 J=JMIN+2,M,1
      Y(J)=Y(J-1)*(2+FAC*H*H*(U(J-1)-E)/
     *    (1-FAC*H*H*(U(J-1)-E)/12))-Y(J-2)
   60 CONTINUE
C
      DO 70 J=JMIN+1,M,1
      Y(J)=Y(J)/Y(M)
   70 CONTINUE
C
C     CALCULATON OF NEW ENERGY EIGENVALUE
C     ***********************************
C
      SOM=0
      DO 80 J=JMIN+1,JMAX-1,1
      SOM=SOM+(Y(J)/(1-FAC*H*H*(U(J)-E)/12))**2
   80 CONTINUE
```

```
      DE=((-Y(M-1)+2*Y(M)-Y(M+1))/H/H/FAC+U(M)-E)/SOM
      E=E+DE
      IT=IT+1
      IF (TERM.EQ.1) WRITE(3,18) IT,M,R(M),E,DE
   18 FORMAT(´ ´,I3,´  ´,I4,´  ´,E14.7,´  ´,E14.7,´  ´,E14.7)
C
      IF(ABS(DE).GT.EPS) GOTO 130
C
C     DETERMINATION OF VIBRATIONAL QUANTUM NUMBER
C     *******************************************
C
      NO=0
      INO=0
      DO 90 IO=JMIN+1,JMAX
      IF(ABS(Y(IO)).LE.ABS(Y(IO-1)).AND.INO.EQ.0) NO=NO+1
      IF(ABS(Y(IO)).LE.ABS(Y(IO-1)).AND.INO.EQ.0) INO=1
      IF(ABS(Y(IO)).GT.ABS(Y(IO-1)).AND.INO.EQ.1) INO=0
   90 CONTINUE
      IF(TERM.EQ.1) WRITE(3,19) NO
   19 FORMAT(/,´ STOPCRITERIUM BEREIKT, VIBR. QUANTUMGETAL= ´,I4,/)
C  19 FORMAT(/,´ STOPCRITERIUM REACHED, VIBR. QUANTUM NUMBER= ´,I4,/)
      IF(VMU.EQ.1) GOTO 95
C
      IF(NO.EQ.VIB+1) GOTO 95
      IF(NO.GT.VIB+1.AND.VIB.GT.0) E=E+EU(VIBC)/2
      IF(NO.GT.VIB+1.AND.VIB.EQ.0) E=E-OMEU/4
      IF(NO.LT.VIB+1) E=E+OMEU
      IF(E.GT.EMAX.AND.MODE.EQ.2) GOTO 140
C
C     NORMALIZATION AND STORAGE OF CALCULATED WAVE-FUNCTION
C     *****************************************************
C
   95 IF(VIBC.GT.0) OMEU=(E-EU(VIBC))/STEP
      VIB=VIB+STEP
      VIBC=VIBC+1
      EU(VIBC)=E
C
      DO 125 I=1,200
      IF(I.NE.VIBC) GOTO 125
      DO 100 J=1,JMIN,1
      PSI(I,J)=0.0
  100 CONTINUE
C
      DO 110 J=JMIN+1,JMAX,1
      PSI(I,J)=Y(J)/(1-FAC*H*H*(U(J)-E)/12)/SQRT(SOM)*SQRT((N+1)*H)
  110 CONTINUE
C
      DO 120 J=JMAX+1,N+1,1
      PSI(I,J)=0.0
  120 CONTINUE
  125 CONTINUE
C
      IF(VMU.EQ.1) GOTO 140
      E=E+OMEU*STEP
      IF(VIB.GE.VMU.AND.MODE.EQ.1) GOTO 140
      IF(E.GT.EMAX.AND.MODE.EQ.2) GOTO 140
C
      IT=0
      GOTO 5
C
  130 IF(IT.LT.ITMAX) GOTO 25
C
```

```
  140 RETURN
      END
C
C
C
      SUBROUTINE VARCOM(RMIN,RMAX,N,VMU,STEP,NAC,STEP2,
     *    UPPARR,LOWARR,GAMMA,C1,C2,C3,C4,C5,MURED)
C
C     This subroutine is used to create a printerfile containing
C     information on the parameters used in the calculation.
C     All parameters have the meaning introduced in the main source
C     program
C
      REAL*4 RMIN,RMAX,GAMMA,MURED,UPPARR,LOWARR,C1,C2,C3,C4,C5
      INTEGER N,VMU,STEP,NAC,STEP2
      DIMENSION UPPARR(9),LOWARR(9)
C
      WRITE(3,1) RMIN,RMAX,N,VMU,STEP,NAC,STEP2
    1 FORMAT(/,´ A) OPERATIE-VARIABELEN´,//,´ RMIN= ´,F9.2,´ RMAX= ´,
     *    F9.2,/,´ AANTAL PUNTEN= ´,I4,//,
     *    ´ AANTAL GEBONDEN TOESTANDEN= ´,I4,/,
     *    ´ INTERVAL VIBRATIONEEL QUANTUMGETAL= ´,I4,/,
     *    ´ AANTAL VRIJE TOESTANDEN= ´,I4,/,
     *    ´ ENERGIE-INCREMENT VRIJE TOESTANDEN= ´,I4,´ CM-1´,///)
C 1   FORMAT(/, A) OPERATION-VARIABLES´,//,´ RMIN= ´,F9.2,´ RMAX= ´,
C    *    F9.2,/,´ AMOUNT OF STEPS= ´,I4,//,
C    *    ´ AMOUNT OF BOUND STATES= ´,I4,/,
C    *    ´ INTERVAL VIBRATIONAL QUANTUM NUMBER= ´,I4,/,
C    *    ´ AMOUNT OF FREE STATES= ´,I4,/,
C    *    ´ ENERGY INCREMENT OF FREE STATES= ´,I4,´ CM-1´,///)
C
      WRITE(3,2)
    2 FORMAT(´ B) SYSTEEM-VARIABELEN´,//,
     *    ´ PARAMETERS VOOR GEBONDEN EN VRIJE TOESTANDEN:´,/)
C 2   FORMAT(´ B) SYSTEM VARIABLES´,//,
C    *    ´ PARAMETERS FOR BOUND AND FREE STATES:´,/)
      DO 10 I=1,9
      WRITE(3,3) I,UPPARR(I),LOWARR(I)
    3 FORMAT(´ I= ´,I2,´  UPPARR(I)= ´,E14.7,´  LOWARR(I)= ´,E14.7)
   10 CONTINUE
      WRITE(3,4)
    4 FORMAT(/,´ PARAMETERS VOOR HET ELECTRONISCH OVERGANGSMOMENT´,/)
C 4   FORMAT(/,´ PARAMETERS FOR THE ELECTRONIC TRANSITION MOMENT´,/)
      WRITE(3,5) GAMMA,C1,C2,C3,C4,C5
    5 FORMAT(´ GAMMA= ´,E14.7,/,´ C1= ´,E14.7,/,´ C2= ´,E14.7,/,
     *    ´ C3= ´,E14.7,/,´ C4= ´,E14.7,/,´ C5= ´,E14.7,/)
C*
      RETURN
      END
```

MINIB2/S/GEBONDENVRIJ/BIJEENVEEG
DATE & TIME PRINTED: MONDAY, JULY 20, 1987 @ 10:47:18.

```
C     MINIB2/S/GEBONDENVRIJ/BIJEENVEEG
C
C     This program is part of a series of programs to simulate
C     bound-free rare gas halide emission spectra.
C     This program uses spectra stored on a specified FILE 1, and
C     copies them to a specified FILE 2, where however spectra from
C     a number of vibrational states are collected into one function.
C     This is a very useful, I/O time-saving preparation for later
C     use of the calculated spectra in programs MINIB2/S/GEBONDENVRIJ/
C     TRANSAANPASSING and MINIB2/S/GEBONDENVRIJ/AANPASSINGCA, where
C     a procedure of this kind would have to be followed anyway.
C
C*******************************************************************
C
FILE  1(KIND=DISK,TITLE="MINIB2/D/GEBONDENVRIJ/XEBRCA/2")
FILE  2(KIND=DISK,TITLE="MINIB2/D/GEBONDENVRIJ/XEBRCA/3",
     *   NEWFILE=TRUE,PROTECTION=SAVE)
FILE  3(KIND=REMOTE)
      INTEGER LMIN,LMAX,VIB,STEP,VIB2,STEP2
      REAL STEP3,UMINUP,EU,EU2,SPEK,SPEK2
      DIMENSION SPEK(0:179,0:1019),SPEK2(0:14,0:1019),EU(0:179),
     *   EU2(0:179)
C
      READ(1,/) LMIN,LMAX,STEP3
      READ(1,/) VIB,STEP
      READ(1,/) UMINUP
      READ(1,/) EU
C
      VIB2=10
      STEP2=15
      IF(VIB2*STEP2.GT.VIB) WRITE(3,5)
      IF(VIB2*STEP2.GT.VIB) GOTO 50
    5 FORMAT(´ OP FILE 1 STAAN ONVOLDOENDE BEREKENDE SPECTRA.´)
C 5   FORMAT(´ INSUFFICIENT AMOUNT OF SPECTRA ON FILE 1.´)
C
C     The parameter STEP2 determines the amount of spectra of
C     individual vibrational states which have to be collected into
C     one fitfunction. The parameter VIB2 determines the amount of
C     fitfunction which are to be formed in this way.
C
      WRITE(2,/) LMIN,LMAX,STEP3
      WRITE(2,/) VIB2,1
      WRITE(2,/) UMINUP
      DO 20 I=0,VIB-1
      DO 10 J=1,34
      READ(1) (SPEK(I,30*(J-1)+K),K=0,29)
   10 CONTINUE
   20 CONTINUE
C
      DO 25 I=0,VIB2-1
      DO 22 K=0,STEP2-1
      EU2(I)=EU2(I)+EU(STEP2*I+K)/STEP2
      DO 30 J=0,1019
   30 SPEK2(I,J)=SPEK2(I,J)+SPEK(STEP2*I+K,J)
   22 CONTINUE
   25 CONTINUE
C
      WRITE(2,/) EU2
```

```
      DO 40 I=0,VIB2-1
      DO 35 J=1,34
      WRITE(2) (SPEK2(I,30*(J-1)+K),K=0,29)
   35 CONTINUE
   40 CONTINUE
C
   50 CLOSE(2,DISP=CRUNCH)
      END
```

```
MINIB2/S/GEBONDENVRIJ/AANPASSINGBX
DATE & TIME PRINTED: MONDAY, JULY 20, 1987 @ 10:47:36.
```

```
C       MINIB2/S/GEBONDENVRIJ/AANPASSINGBX
C
C       This program is part of a series of programs to simulate
C       bound-free rare gas halide emission spectra.
C       When doing simulations of experimental emission spectra it
C       often occurs that, having calculated the emission spectra of
C       individual vibrational states for a given set of potential
C       curves and a given transition moment, one wants to modify
C       either of these. For this purpose there are two programs,
C       the present one, to be used for transitions where there may
C       be more than one point of stationary phase for a given combi-
C       nation of upper state and lower state energies (for example
C       B-X(1/2) transitions), and MINIB2/S/GEBONDENVRIJ/AANPASSINGCA,
C       when there is only one point of stationary phase (for example
C       C-A(3/2) transitions). Since there has as yet never been the
C       necessity of transforming B-X(1/2) potential curves, the present
C       program only has the option of changing the transition moment.
C       The structure of the program is such that spectra on a
C       specified FILE 1 are transformed and copied to a specified
C       FILE 2.
C
C*******************************************************************
C
C       DECLARATIONS
C       ************
C
FILE  1(KIND=DISK,FILETYPE=7,TITLE="MINIB2/D/GEBONDENVRIJ/XEBRBX/4")
FILE  2(KIND=DISK,TITLE="MINIB2/D/GEBONDENVRIJ/XEBRBX/5",
     *  PROTECTION=SAVE,NEWFILE=TRUE)
FILE  3(KIND=REMOTE)
FILE  4(KIND=DISK, FILETYPE=7, TITLE="MINIB2/D/GEBONDENVRIJ/POTINVOER")
FILE  5(KIND=PRINTER)
        REAL STEP3,UMINUP,EU,SPEK,DUMMY,RMIN,RMAX,UPPARR,
     *     LOWARR,H,UDIF,R,UUP,ULO,UMUL,REUP,GAMMA,GAMMA2,
     *     D1,D2,D3,D4,D5,C1,C2,C3,C4,C5,TM,TM2,RBI,LABBI1,LABBI2,
     *     LABMIN,RBU,LABMAX,G,LAB,NU,R1,
     *     R2
        INTEGER STEP,LMIN,LMAX,VIB,N,IREUP,IBI,IBU,I1,I2
        DIMENSION EU(0:179),SPEK(0:179,0:1019),UPPARR(1:9),
     *     LOWARR(1:9),UUP(0:1001),ULO(0:1001),UMUL(0:1001),
     *     TM(0:1001),TM2(0:1001),G(0:1001),R(0:1001),
     *     IR1(0:1001),IR2(0:1001),IBI(0:179),IBU(0:179)
C
C       INPUT VARIABLES
C       ***************
C
        READ(1,/) LMIN,LMAX,STEP3
        READ(1,/) VIB,STEP
        READ(1,/) UMINUP
        READ(1,/) EU
        WRITE(2,/) LMIN,LMAX,STEP3
        WRITE(2,/) VIB,STEP
        WRITE(2,/) UMINUP
        WRITE(2,/) EU
C
        DO 40 I=0,VIB-1
        DO 30 J=1,34
        READ(1) (SPEK(I,30*(J-1)+K),K=0,29)
```

```
   30 CONTINUE
   40 CONTINUE
C
      READ(4,/) DUMMY
      READ(4,/) DUMMY
      READ(4,/) RMIN,RMAX
      READ(4,/) N
      READ(4,/) DUMMY,DUMMY
      READ(4,/) DUMMY,DUMMY
      READ(4,/) DUMMY,DUMMY
      READ(4,/) DUMMY,DUMMY
      READ(4,/) DUMMY
      READ(4,/) UPPARR(1),
     *   UPPARR(2),UPPARR(3),UPPARR(4),UPPARR(5),UPPARR(6),
     *   UPPARR(7),UPPARR(8),UPPARR(9)
      READ(4,/) LOWARR(1),
     *   LOWARR(2),LOWARR(3),LOWARR(4),LOWARR(5),LOWARR(6),
     *   LOWARR(7),LOWARR(8),LOWARR(9)
C
C     CALCULATION OF POTENTIAL CURVES
C     *******************************
C
      H=(RMAX-RMIN)/(N+1)
      UMINUP=10**10
      UDIF=10**10
C
      DO 31 I=0,N+1,1
      R(I)=RMIN+I*H
C     R(I)=R(I)*1.09        Note: for XeBr B-X(1/2_ calculation!
      UUP(I)=UPPARR(1)+UPPARR(2)
     *   +UPPARR(3)*EXP(-UPPARR(4)*R(I))
     *   +UPPARR(5)*EXP(-UPPARR(6)*R(I))
     *   -UPPARR(7)/R(I)
     *   -UPPARR(8)/(R(I)**4)
     *   -UPPARR(9)/(R(I)**6)
      ULO(I)=LOWARR(1)+LOWARR(2)
     *   +LOWARR(3)*EXP(-LOWARR(4)*R(I))
     *   +LOWARR(5)*EXP(-LOWARR(6)*R(I))
     *   -LOWARR(7)/R(I)
     *   -LOWARR(8)/(R(I)**4)
     *   -LOWARR(9)/(R(I)**6)
      UMUL(I)=UUP(I)-ULO(I)
      IF(UMUL(I).LE.UDIF) IMUL=I
      IF(UMUL(I).LE.UDIF) UDIF=UMUL(I)
      IF(UUP(I).GE.UMINUP) GOTO 37
      UMINUP=UUP(I)
      IREUP=I
      REUP=R(I)
   37 CONTINUE
C     R(I)=R(I)/1.09        Note: for XeBr B-X(1/2) calculation!
   31 CONTINUE
C
C     PARAMETERS FOR THE OLD TRANSITION MOMENT
C     ***************************************
C
      GAMMA2=0.3
      D1=33.50
      D2=-36.25
      D3=14.70
      D4=-2.648
      D5=0.182
C
```

```
      READ(4,/) DUMMY
      READ(4,/) GAMMA,C1,C2,C3,C4,C5
C
C     Parameters for the new transition moment are stored on
C     FILE 4, the inputfile for MINIB2/S/GEBONDENVRIJ/POTCALC.
C
C     CALCULATION OF THE OLD AND NEW TRANSITION MOMENTS
C     ***********************************************
C
      DO 41 I=0,N+1
C     R(I)=R(I)*1.09        Note: for XeBr B-X(1/2) calculation!
      TM(I)=EXP(0.6*(R(I)-REUP))/
     *   (C1+C2*R(I)+C3*R(I)**2+C4*R(I)**3+C5*R(I)**4)
C    *   EXP(0.2*((R(I)-REUP)**2))
C     R(I)=R(I)/1.09        Note: for XeBr B-X(1/2) calculation!
      TM2(I)=EXP(GAMMA2*(R(I)-REUP))/
     *   (D1+D2*R(I)+D3*R(I)**2+D4*R(I)**3+D5*R(I)**4)
C
C     The arrays TM and TM2 are used to store the new and the old
C     transition moment, respectively.
C
   41 CONTINUE
C
C     DETERMINATION OF INNER AND OUTER TURNING POINTS OF THE BOUND
C     UPPER STATES
C     ***********************************************************
C
      DO 100 I=0,VIB-1
      IREUP2=IREUP
      CALL SNIJPU(UUP,0,IREUP2,EU(I),RBI)
C
C     Subroutine SNIJPU is a subroutine which calculates, within
C     an interval determined by array-indices I1 and I2, the
C     location where a function F takes on a specified value E.
C     See source text of SNIJPU.
C
      IBI(I)=INT(RBI)+1
      IREUP2=IREUP
      N2=N
      CALL SNIJPU(UUP,IREUP2,N2,EU(I),RBU)
      IBU(I)=INT(RBU)
  100 CONTINUE
C
      DO 80 J=0,N
      G(J)=ABS(1/(UUP(J+1)-UUP(J)-ULO(J+1)+ULO(J)))
C
C     Array G is used to store the inverse of the difference
C     between the gradient of the upper and lower state potential
C     curves. G(J) is a measure for the width of the stationary
C     phase contributions around R(J).
C
   80 CONTINUE
C
C     TRANSFORMATION OF CALCULATED EMISSION SPECTRA
C     *********************************************
C
      LABMIN=10**7/(EU(VIB-1)-LOWARR(1))
      LABMAX=10**7/UDIF
      JMIN=INT((LABMIN-LMIN)/STEP3)+1
      JMAX=INT((LABMAX-LMIN)/STEP3)
C
C     Determination of the location of the points of stationary
```

```
C     phase as a function of the wavelength of the emitted photon.
C
      DO 90 J=JMIN,JMAX
      LAB=LMIN+J*STEP3
      NU=10**7/LAB
      IMUL2=IMUL
      CALL SNIJPU(UMUL,0,IMUL2,NU,R1)
      IR1(J)=INT(R1)
      IMUL2=IMUL
      N2=N
      CALL SNIJPU(UMUL,IMUL2,N2,NU,R2)
      IR2(J)=INT(R2)
   90 CONTINUE
C
      DO 95 I=0,VIB-1
C
C     Selection of points of stationary phase which are classically
C     attainable, followed by transformation of the calculated
C     emission spectra
C
      DO 94 J=JMIN,JMAX
      IF(IR1(J).LT.IBI(I).AND.IR2(J).GT.IBU(I)) GOTO 94
      IF(IR1(J).GE.IBI(I).AND.IR2(J).LE.IBU(I)) GOTO 92
      IF(IR1(J).GE.IBI(I).AND.IR2(J).GT.IBU(I)) GOTO 91
      SPEK(I,J)=SPEK(I,J)+(TM(IR2(J))/TM2(IR2(J)))**2
      GOTO 94
   91 SPEK(I,J)=SPEK(I,J)*(TM(IR1(J))/TM2(IR1(J)))**2
      GOTO 94
   92 F1=1/SQRT(EU(I)-UUP(IR1(J)))*G(IR1(J))
      F2=1/SQRT(EU(I)-UUP(IR2(J)))*G(IR2(J))
      SPEK(I,J)=SPEK(I,J)*
     *   (F1*TM(IR1(J))**2+F2*TM(IR2(J))**2)/
     *   (F1*TM2(IR1(J))**2+F2*TM2(IR2(J))**2)
   94 CONTINUE
   95 CONTINUE
C
      DO 140 I=0,VIB-1
      DO 130 J=1,34
      WRITE(2) (SPEK(I,30*(J-1)+K),K=0,29)
  130 CONTINUE
  140 CONTINUE
C
      CLOSE(6,DISP=CRUNCH)
      END
C
C
      SUBROUTINE SNIJPU(F,I1,I2,E,R3)
C
C     Subroutine SNIJPU is a subroutine which calculates, within
C     boundaries determined by the array-indices I1 and I2, the
C     location where a function F takes on a specified value E.
C     As output the subroutine gives a parameter R3 which
C     is an optimal real approximation of the array-index where
C     the function F takes on the value E. If there is no point
C     between I1 and I2 where F takes on the value E, then R3 is
C     given the value -1.
C
      REAL F,E,R3
      INTEGER I1,I2,I3
      DIMENSION F(0:4001)
C*
      IF((I2-I1).LE.1) GOTO 30
```

```
      IF(F(I1).LT.E.AND.F(I2).LT.E) R3=-1
      IF(F(I1).GE.E.AND.F(I2).GE.E) R3=-1
      IF(R3.LT.0) GOTO 40
C*
      IF(F(I2).GE.F(I1)) GOTO 20
   10 I3=INT((I2+I1)/2)
      IF(F(I3).LT.E) I2=I3
      IF(F(I3).GT.E) I1=I3
      IF((I2-I1).LE.1) GOTO 30
      GOTO 10
C*
   20 I3=INT((I2+I1)/2)
      IF(F(I3).LT.E) I1=I3
      IF(F(I3).GE.E) I2=I3
      IF((I2-I1).LE.1) GOTO 30
      GOTO 20
C
   30 R3=I1+(F(I1)-E)/(F(I1)-F(I2))
   40 CONTINUE
C
      RETURN
      END
```

MINIB2/S/GEBONDENVRIJ/AANPASSINGCA
DATE & TIME PRINTED: MONDAY, JULY 20, 1987 @ 10:47:53.

```
C     MINIB2/S/GEBONDENVRIJ/AANPASSINGCA
C
C     This program is part of a series of programs to simulate
C     bound-free rare gas halide emission spectra
C     When doing simulations of experimental emission spectra it
C     often occurs that, having calculated the emission spectra of
C     individual vibrational states for a given set of potential
C     curves and a given transition moment, one wants to modify
C     either of these. For this purpose there are two programs,
C     MINIB2/S/GEBONDENVRIJ/AANPASSINGBX, to be used for transtions
C     where there may be more than one point of stationary phase
C     for a given combination of upper state and lower state
C     energies (for example B-X(1/2) transitions), and the present
C     one, to be used when there is only one point of stationary
C     phase (for example C-A(3/2 transitions).
C
C**********************************************************************
C
C     DECLARATIONS
C     ************
C
FILE  1(KIND=DISK,FILETYPE=7,TITLE="MINIB2/D/GEBONDENVRIJ/XEBRCA/3")
FILE  2(KIND=DISK,TITLE="MINIB2/D/GEBONDENVRIJ/XEBRCA/4",
     *  PROTECTION=SAVE,NEWFILE=TRUE)
FILE  3(KIND=REMOTE)
FILE  4(KIND=DISK,FILETYPE=7,TITLE="MINIB2/D/GEBONDENVRIJ/POTINVOERCA")
FILE  5(KIND=PRINTER)
      REAL STEP3,UMINUP,EU,SPEK,DUMMY,RMIN,RMAX,UPPARR,
     *   LOWARR,H,UDIF,R,UUP,ULO,ULO2,UMUL,UMUL2,REUP,GAMMA,GAMMA2,
     *   D1,D2,D3,D4,D5,C1,C2,C3,C4,C5,TM,TM2,RBI,LABBI1,LABBI2,
     *   LABMIN,RBU,LABMAX,G,LAB,NU,R1,LABIAC,LABUAC,RKOUD,
     *   R2,DELTTE,CORR,LABBI,LABBU,LABOUD,ROUD,RNU,NUOUD,LOUD
      INTEGER LMIN,LMAX,KMAX,VIB,N,IREUP,IBI,IBU,I1,I2,JBIN,
     *   JBUI,IOUD,SPEK2,INU,IKOUD,STEP
      DIMENSION EU(0:179),SPEK(0:29,0:1019),UPPARR(1:9),
     *   LOWARR(1:9),UUP(0:1001),ULO(0:1001),UMUL(0:1001),
     *   TM(0:1001),TM2(0:1001),G(0:1001),R(0:1001),
     *   IR1(0:1001),IR2(0:1001),IBI(0:179),IBU(0:179),
     *   SPEK2(0:29,0:1019),UMUL2(0:1001),ULO2(0:1001),
     *   RULO(0:9),UULO(0:9)
C
C     INPUT VARIABLES
C     ***************
C
      READ(1,/) LMIN,LMAX,STEP3
      READ(1,/) VIB,STEP
      READ(1,/) UMINUP
      READ(1,/) EU
C
      WRITE(2,/) LMIN,LMAX,STEP3
      WRITE(2,/) VIB,STEP
      WRITE(2,/) UMINUP
      WRITE(2,/) EU
C
      DO 40 I=0,VIB-1
      DO 30 J=1,34
      READ(1) (SPEK(I,30*(J-1)+K),K=0,29)
   30 CONTINUE
```

```
  40  CONTINUE
C
      READ(4,/) DUMMY
      READ(4,/) DUMMY
      READ(4,/) RMIN,RMAX
      READ(4,/) N
      READ(4,/) DUMMY,DUMMY
      READ(4,/) DUMMY,DUMMY
      READ(4,/) DUMMY,DUMMY
      READ(4,/) DUMMY,DUMMY
      READ(4,/) DUMMY
      READ(4,/) UPPARR(1),
     *   UPPARR(2),UPPARR(3),UPPARR(4),UPPARR(5),UPPARR(6),
     *   UPPARR(7),UPPARR(8),UPPARR(9)
      READ(4,/) LOWARR(1),
     *   LOWARR(2),LOWARR(3),LOWARR(4),LOWARR(5),LOWARR(6),
     *   LOWARR(7),LOWARR(8),LOWARR(9)
C
C     CALCULATION OF POTENTIAL CURVES
C     ******************************
C
      H=(RMAX-RMIN)/(N+1)
      UMINUP=10**10
      UDIF=10**10
C
      DO 31 I=0,N+1,1
      R(I)=RMIN+I*H
      UUP(I)=UPPARR(1)+UPPARR(2)
     *   +UPPARR(3)*EXP(-UPPARR(4)*R(I))
     *   +UPPARR(5)*EXP(-UPPARR(6)*R(I))
     *   -UPPARR(7)/R(I)
     *   -UPPARR(8)/(R(I)**4)
     *   -UPPARR(9)/(R(I)**6)
      ULO(I)=LOWARR(1)+LOWARR(2)
     *   +LOWARR(3)*EXP(-LOWARR(4)*R(I))
     *   +LOWARR(5)*EXP(-LOWARR(6)*R(I))
     *   -LOWARR(7)/R(I)
     *   -LOWARR(8)/(R(I)**4)
     *   -LOWARR(9)/(R(I)**6)
      ULO2(I)=4.8536E7*EXP(-2.9464*R(I))+600
C
C     A modification of the potential curves is achieved by
C     introducing a new lower state potential curve ULO2.
C
      UMUL(I)=UUP(I)-ULO(I)
      UMUL2(I)=UUP(I)-ULO2(I)
      IF(10*(INT(I/10)).EQ.I) WRITE(5,/) R(I),UUP(I),ULO(I),ULO2(I),
     *   UMUL2(I)
      IF(UMUL(I).LE.UDIF) IMUL=I
      IF(UMUL(I).LE.UDIF) UDIF=UMUL(I)
      IF(UUP(I).GE.UMINUP) GOTO 37
      UMINUP=UUP(I)
      IREUP=I
      REUP=R(I)
  37  CONTINUE
  31  CONTINUE
C
C     PARAMETERS FOR THE OLD TRANSITION MOMENT
C     ****************************************
C
      GAMMA2=0.0
      D1=19.96
```

```
      D2=-25.62
      D3=12.57
      D4=-2.799
      D5=0.248
C
      READ(4,/) DUMMY
      READ(4,/) GAMMA,C1,C2,C3,C4,C5
C
C     Parameters for the new transition moment are stored on
C     FILE 4, the inputfile for MINIB2/S/GEBONDENVRIJ/POTCALC
C
C     CALCULATION OF THE OLD AND NEW TRANSITION MOMENTS
C     ***********************************************
C
      DO 41 I=0,N+1
C     GAMMA=0.4       For quick changes FILE 4 may be overruled.
      TM(I)=EXP(GAMMA*(R(I)-REUP))/
     *   (C1+C2*R(I)+C3*R(I)**2+C4*R(I)**3+C5*R(I)**4)
      TM2(I)=EXP(GAMMA2*(R(I)-REUP))/
     *   (D1+D2*R(I)+D3*R(I)**2+D4*R(I)**3+D5*R(I)**4)
C
C     The arrays TM and TM2 are used to store the new and the old
C     transition moment, respectively.
C
   41 CONTINUE
C
C     TRANSFORMATION OF THE TRANSITION MOMENT OF CALCULATED EMISSION
C     SPECTRA
C     **************************************************************
C
      LABMIN=10**7/(EU(VIB-1)-LOWARR(1))
      LABMAX=LMAX
      JMIN=INT((LABMIN-LMIN)/STEP3)+1
      JMAX=INT((LABMAX-LMIN)/STEP3)
C
C     Determination of the location of the points of stationary
C     phase as a function of the wavelength of the emitted photon.
C
      DO 90 J=JMIN,JMAX
      LAB=LMIN+J*STEP3
      NU=10**7/LAB
      IMUL2=IMUL
      N2=N
      CALL SNIJPU(UMUL,0,N2,NU,R1)
      IR1(J)=INT(R1)
   90 CONTINUE
C
C     Transformation of the calculated emission spectra.
C
      DO 95 I=0,VIB-1
      DO 94 J=JMIN,JMAX
      IF(IR1(J).EQ.-1) GOTO 94
   91 SPEK(I,J)=SPEK(I,J)*(TM(IR1(J))/TM2(IR1(J)))**2
   94 CONTINUE
   95 CONTINUE
C
C     TRANSFORMATION OF THE POTENTIAL CURVES OF CALCULATED EMISSION
C     SPECTRA
C     *************************************************************
C
C     Determination of internuclear distance for the new potential
C     curves where an emitted photon will have wavelength LAB.
```

```
C
      KMAX=INT((LMAX-LMIN)/STEP3)
      DO 200 K=0,KMAX
      LAB=LMIN+K*STEP3
      NU=1E7/LAB
      RNU=-1
      IF(RNU.NE.-1) GOTO 210
      DO 210 J=0,N
      IF(UMUL2(J).LE.NU.AND.UMUL2(J+1).GT.NU) RNU=(NU-
     *   UMUL2(J))/(UMUL2(J+1)-UMUL2(J))+J
   210 CONTINUE
      INU=INT(RNU)
C
C     Calculation of the wavelength of an emitted photon for the
C     old potential curves, at the previously determined internuclear
C     distance.
C
      IF(INU.EQ.-1) GOTO 200
      NUOUD=UMUL(INU)*(1-RNU+INU)+UMUL(INU+1)*(RNU-INU)
      LOUD=1E7/NUOUD
C
C     Transformation of the calculated emission spectrum.
C
      RKOUD=(LOUD-LMIN)/STEP3
      IKOUD=INT(RKOUD)
      IF(IKOUD.LT.0.OR.IKOUD.GT.1018) GOTO 220
      DO 220 I=0,VIB-1
      SPEK2(I,K)=SPEK(I,IKOUD)*(1-RKOUD+IKOUD)+
     *   SPEK(I,IKOUD+1)*(RKOUD-IKOUD)
  220 CONTINUE
C
  200 CONTINUE
C
      DO 140 I=0,VIB-1
      DO 130 J=1,34
      WRITE(2) (SPEK2(I,30*(J-1)+K),K=0,29)
  130 CONTINUE
  140 CONTINUE
C
      CLOSE(2,DISP=CRUNCH)
      END
C
C
      SUBROUTINE SNIJPU(F,I1,I2,E,R3)
C
C     For explanation see MINIB2/S/GEBONDENVRIJ/AANPASSINGBX.
C
      REAL F,E,R3
      INTEGER I1,I2,I3
      DIMENSION F(0:4001)
C
      IF((I2-I1).LE.1) GOTO 30
      IF(F(I1).LT.E.AND.F(I2).LT.E) R3=-1
      IF(F(I1).GE.E.AND.F(I2).GE.E) R3=-1
      IF(R3.LT.0) GOTO 40
C
      IF(F(I2).GE.F(I1)) GOTO 20
  10  I3=INT((I2+I1)/2)
      IF(F(I3).LT.E) I2=I3
      IF(F(I3).GT.E) I1=I3
      IF((I2-I1).LE.1) GOTO 30
      GOTO 10
```

```
C*
   20 I3=INT((I2+I1)/2)
      IF(F(I3).LT.E) I1=I3
      IF(F(I3).GE.E) I2=I3
      IF((I2-I1).LE.1) GOTO 30
      GOTO 20
C
   30 R3=I1+(F(I1)-E)/(F(I1)-F(I2))
   40 CONTINUE
C
      RETURN
      END
```

MINIB2/S/GEBONDENVRIJ/GLADSTRIJKEN
DATE & TIME PRINTED: MONDAY, JULY 20, 1987 @ 10:48:14.

```
C     MINIB2/S/GEBONDENVRIJ/GLADSTRIJKEN
C
C     This program is part of a series of programs to simulate
C     bound-free rare gas halide emission spectra.
C     With this program calculated C-A(3/2) emission spectra from
C     individual vibrational states can be given an envelope which
C     goes down exponentially towards shorter wavelengths. This is
C     achieved by fitting the calculated secondary maxima to an
C     exponential function.
C
C*****************************************************************
C
FILE 1(KIND=DISK,TITLE="MINIB2/D/GEBONDENVRIJ/XEBRCA/4")
FILE 3(KIND=PRINTER)
FILE 2(KIND=DISK,TITLE="MINIB2/D/GEBONDENVRIJ/XEBRCA/5",
     *   NEWFILE=TRUE, PROTECTION=SAVE)
      REAL EU,SPEK,SPEK2,STEP3,UMINUP,LAB,INTENS,INTL,INTI,
     *   INTLI,INTLKW,B,A,CORR1,CORR2,LNA,K1,K2
      INTEGER LMIN,LMAX,VIB,STEP,INDEX,LABJ
      DIMENSION SPEK(0:9,0:1019),SPEK2(0:9,0:1019),EU(0:179),
     *   LABJ(1:100),LAB(1:100),INTENS(1:100)
C
      READ(1,/) LMIN,LMAX,STEP3
      READ(1,/) VIB,STEP
      READ(1,/) UMINUP
      READ(1,/) EU
C
      WRITE(2,/) LMIN,LMAX,STEP3
      WRITE(2,/) VIB,STEP
      WRITE(2,/) UMINUP
      WRITE(2,/) EU
C
      DO 20 I=0,9
      DO 10 J=1,34
      READ(1) (SPEK(I,30*(J-1)+K),K=0,29)
  10  CONTINUE
  20  CONTINUE
C
      DO 50 I=2,VIB-1
C
      SPEKMA=0
      DO 51 J=0,1019
      IF(SPEK(I,J).GT.SPEKMA) JMAX=J
      IF(SPEK(I,J).GT.SPEKMA) SPEKMA=SPEK(I,J)
  51  CONTINUE
C
C     Determination of the location of the secondary maxima
C
      INDEX=0
      DO 40 J=400,JMAX
      IF(SPEK(I,J-3).GT.SPEK(I,J)) GOTO 40
      IF(SPEK(I,J-2).GT.SPEK(I,J)) GOTO 40
      IF(SPEK(I,J-1).GT.SPEK(I,J)) GOTO 40
      IF(SPEK(I,J+1).GE.SPEK(I,J)) GOTO 40
      IF(SPEK(I,J+2).GE.SPEK(I,J)) GOTO 40
      IF(SPEK(I,J+3).GE.SPEK(I,J)) GOTO 40
      INDEX=INDEX+1
      LABJ(INDEX)=J
```

```
      LAB(INDEX)=LMIN+J*STEP3
      INTENS(INDEX)=SPEK(I,J)
C     WRITE(3,/) I,INDEX,LABJ(INDEX),LAB(INDEX),INTENS(INDEX)
   40 CONTINUE
C
C     Fit to an exponential function
C
      INTL=0
      INTI=0
      INTLI=0
      INTLKW=0
      IF(INDEX.LE.1) GOTO 50
C
      DO 60 J=1,INDEX
      INTL=INTL+LAB(J)
      INTI=INTI+ALOG(INTENS(J))
      INTLI=INTLI+LAB(J)*ALOG(INTENS(J))
      INTLKW=INTLKW+LAB(J)**2
   60 CONTINUE
      B=(-INTL*INTI+INDEX*INTLI)/(INDEX*INTLKW-INTL**2)
      LNA=((INTLKW*INTI-INTL*INTLI)/(INDEX*INTLKW-INTL**2))
      A=EXP(LNA)
C
C     Modification of the emission spectra
C
      DO 80 J=1,INDEX-1
      K1=LABJ(J)
      K2=LABJ(J+1)
      CORR1=INTENS(J)/A/EXP(B*LAB(J))
      CORR2=INTENS(J+1)/A/EXP(B*LAB(J+1))
      DO 70 K=K1,K2-1
      SPEK(I,K)=SPEK(I,K)*((K-K1)/(K2-K1)/CORR2+(K2-K)/(K2-K1)/CORR1)
   70 CONTINUE
   80 CONTINUE
      DO 65 K=0,LABJ(1)-1
      SPEK(I,K)=SPEK(I,K)*A*EXP(B*LAB(1))/SPEK(I,LABJ(1))
   65 CONTINUE
      DO 66 K=LABJ(INDEX),1019
      SPEK(I,K)=SPEK(I,K)*A*EXP(B*LAB(INDEX))/SPEK(I,LABJ(INDEX))
   66 CONTINUE
C     WRITE(3,110)
C 110 FORMAT(/)
   50 CONTINUE
C
      DO 100 I=0,9
      DO 90 J=1,34
      WRITE(2) (SPEK(I,30*(J-1)+K),K=0,29)
   90 CONTINUE
  100 CONTINUE
C
      CLOSE(2,DISP=CRUNCH)
      END
```

MINIB2/S/GEBONDENVRIJ/VIBFIT
DATE & TIME PRINTED: MONDAY, JULY 20, 1987 @ 10:48:30.

```
C     MINIB2/S/GEBONDENVRIJ/VIBFIT
C
C     This program is part of a series of programs to simulate
C     bound-free rare gas halide emission spectra.
C     In this program an experimental emission spectrum from a
C     single bound-free electronic transition, stored on a specified
C     FILE 2, is simulated using calculated emission spectra of
C     individual vibrational states (or collections of emission
C     spectra of individual vibrational states) stored on a specified
C     FILE 1. Using a least squares method a vibrational distribution
C     of the bound upper state is determined, and stored on a
C     file MINIB2/D/GEBONDENVRIJ/FITUITVOER (FILE 6).
C     The program also creates a previewfile which contains a graphical
C     representation of the experimental emission spectrum and the
C     simulated spectrum.
C
C*******************************************************************
C
C     DECLARATIONS
C     ************
C
FILE  1(KIND=DISK,FILETYPE=7,TITLE="MINIB2/D/GEBONDENVRIJ/XEBRCA/5")
FILE  2(KIND=DISK,FILETYPE=7,TITLE="MINIB2/D/GEBONDENVRIJ/DATAOUT")
FILE  3(KIND=REMOTE)                 % WRITE TO TERMINAL
FILE  4(KIND=REMOTE)                 % READ FROM TERMINAL
FILE  5(KIND=PRINTER)
FILE  6(KIND=DISK,TITLE="MINIB2/D/GEBONDENVRIJ/FITUITVOER",
     *  PROTECTION=SAVE,NEWFILE=TRUE)
FILE  7(KIND=DISK, FILETYPE=7, TITLE="MINIB2/D/GEBONDENVRIJ/POTINVOER")
      BLOCK GLOBALS
 $ INCLUDE "PLOTTER/FORTRAN/DECLARATION ON APPL"
      END
 $ INCLUDE "PLOTTER/FORTRAN/POLYDZ ON APPL"
      REAL*4 STEP3,UMINUP,EU,SPEK,STEP2,DATA,STEPFI,LABDAI,
     *   SPRES,DATA2,RSPKLB,RDATLB,SPEKM,DATAM,UIT,RES,SPEKMA,DATAMA,
     *   CORR,HOR,HOR2,DATAM2,SPEK2,UUP,ULO,R,UPPARR,LOWARR,DUMMY,
     *   RMIN,RMAX,LAB,RBI,LIBI1,LIBI2,LIBI,F,L,FL,LL,LNF,LNA,A,
     *   B,H,DATAR,AGEM,CGEM,LABSP,GEW,SIGMA,LMIN2,LMAX2,UIT3,DATAM3
      INTEGER*4 LMIN,LMAX,VIB,VIBF,STEP,LMIN3,LMAX3,LMINFI,LMAXFI,
     *   NFIT,I,J,ISPKLB,IDATLB,IKAN,NKAN,SBER,NBER,LGRENS,NDATA,
     *   JGRENS
      DIMENSION EU(0:179),SPEK(0:50,0:1019),SPEKM(0:50,0:1019),
     *   DATA(0:1019),DATAM2(0:1019),DATAM(0:1019),UIT(0:180),
     *   SPRES(0:180),HOR(0:1019),UPPARR(1:9),LOWARR(1:9),
     *   HOR2(0:1019),SPEK2(0:50,0:1019),UUP(0:4001),ULO(0:4001),
     *   R(0:4001),LAB(0:1019),DATAR(0:1019),GEW(0:1019),
     *   UIT3(0:180),DATAM3(0:1019)
C
C     INPUT VARIABLES AND INITIALIZATION
C     **********************************
C
      READ(1,/) LMIN,LMAX,STEP3
      READ(1,/) VIBF,STEP
      READ(1,/) UMINUP
      READ(1,/) EU
      WRITE(3,1) VIBF,STEP
    1 FORMAT(´ VIBR. NIVEAUS BEREKEND TOT V=´,I4,/,
     *   ´ STAPGROOTTE=´,I3,/,
```

```
     *    ´ GEEF VBER EN STAPGROOTTE SBER:´)
C 1   FORMAT(´ VIBR. STATES DETERMINED UP TO V=´,I4,/,
C    *    ´ INTERVALSIZE=´,I3,/,
C    *    ´ ENTER VBER AND INTERVALSIZE SBER:´)
  2   READ(4,/) VIB,SBER
      NBER=INT(VIB/SBER)
      IF(VIB/SBER-INT(VIB/SBER).NE.0) GOTO 2
      WRITE(6,/) VIB,SBER
C
      DO 40 I=0,VIB-1
      DO 30 J=1,34
      READ(1) (SPEK(I,30*(J-1)+K),K=0,29)
 30   CONTINUE
 40   CONTINUE
C
      READ(2,/) LMIN3,LMAX3,STEP2
C     LMAX3=310
C
C     If not overruled, the program will try to fit the experimental
C     emission spectrum in the entire wavelength interval determined
C     by LMIN3 and LMAX3.
C
      LMIN2=LMIN3
      LMAX2=LMAX3
C
C     LMIN3 and LMAX3, and LMIN2 and LMAX2, are the integer and real
C     real versions of the wavelength boundaries, respectively.
C
      READ(2,/) DATA
      CORR=-11.9
      LMIN2=LMIN2+CORR
      LMAX2=LMAX2+CORR
C
C     CORR is a correction variable to match the wavelength scales
C     of the experimental and calculated emission spectra.
C
C     Optional: in simulations of B-X(1/2) continua, the influence
C     of overlapping C-A(3/2) (and B-A(1/2)) emission may be
C     accounted for by substracting a straight line contribution.
C
      NDATA=INT((LMAX2-LMIN2)/STEP2)
C     DO 39 J=1,10
C     AGEM=AGEM+DATA(J)/10
C 39  CONTINUE
C     DO 37 J=NDATA-9,NDATA
C     CGEM=CGEM+DATA(J)/10
C 37  CONTINUE
C     LGRENS=225
C     JGRENS=INT((LGRENS-LMIN2)/STEP2)
C     DO 42 J=JGRENS,1019
C     DATA(J)=DATA(J)-(CGEM-AGEM)*(J-JGRENS)/(NDATA-JGRENS)
C 42  CONTINUE
C
C     NORMALIZATION OF CALCULATED AND EXPERIMENTAL EMISSION SPECTRA
C     *************************************************************
C
      SPEKMA=0
      DO 61 I=0,VIB-1
      DO 60 J=0,1019
      IF(SPEK(I,J).GT.SPEKMA) SPEKMA=SPEK(I,J)
 60   CONTINUE
 61   CONTINUE
```

```
C
      DO 63 I=0,VIB-1
      DO 62 J=0,1019
      SPEK(I,J)=SPEK(I,J)/SPEKMA
   62 CONTINUE
   63 CONTINUE
C
      DATAMA=0
      DO 64 J=0,1019
      IF(DATA(J).GT.DATAMA) DATAMA=DATA(J)
   64 CONTINUE
C
      DO 65 J=0,1019
      DATA(J)=DATA(J)/DATAMA*10000
   65 CONTINUE
C
C     DETERMINATION OF WAVELENGTH INTERVAL AND STEPSIZE FOR LEAST
C     SQUARES FIT
C     ***********************************************************
C
      LMINFI=LMIN
      LMAXFI=LMAX
      IF(LMIN2.GT.LMIN) LMINFI=INT(LMIN2+.99)
      IF(LMAX2.LT.LMAX) LMAXFI=LMAX2
      LMINFI=250
C
      STEPFI=STEP3
      IF(STEP2.GT.STEP3) STEPFI=STEP2
      NFIT=INT((LMAXFI-LMINFI)/STEPFI)
C
C     CREATION OF FIT-FUNCTIONS FOR LEAST SQUARES FIT
C     ***********************************************
C
C     a) Joining of SBER emission spectra of individual vibrational
C        states into a single fit-function.
C
      IF(SBER.EQ.1) GOTO 17
      DO 17 I=0,1019
      DO 15 J=0,NBER-1
      SPEK(J,I)=SPEK(SBER*J,I)
      DO 16 K=1,SBER-1
      SPEK(J,I)=SPEK(J,I)+SPEK(SBER*J+K,I)
   16 CONTINUE
      SPEK(J,I)=SPEK(J,I)/SBER
   15 CONTINUE
   17 CONTINUE
      DO 14 I=0,1019
      SPEK(NBER,I)=1
   14 CONTINUE
C
C     b) The introduction of a spectral resolution into the fit-
C        functions
C
      NKAN=0
      DO 35 IKAN=-NKAN,NKAN
      DO 34 J=0,NBER
      DO 33 I=0,1019
      IF(I+IKAN.GE.0.AND.I+IKAN.LE.1019) GOTO 41
      SPEK2(J,I)=SPEK2(J,I)+SPEK(J,I)
      GOTO 32
   41 CONTINUE
      SPEK2(J,I)=SPEK2(J,I)+SPEK(J,I+IKAN)
```

```
   32 CONTINUE
   33 CONTINUE
   34 CONTINUE
   35 CONTINUE
C
C     CONVERSION OF CALCULATED AND EXPERIMENTAL EMISSION SPECTRA
C     TO SPECTRA MATCHING THE REQUIRED WAVELENGTH INTERVAL AND
C     HAVING THE CORRECT STEPSIZE.
C     **************************************************************
C
      DO 20 I=0,NFIT
      LABDAI=LMINFI+I*STEPFI
C
      RSPKLB=((LABDAI-LMIN)/STEP3)
      ISPKLB=INT(RSPKLB)
      DO 10 J=0,NBER
      SPEKM(J,I)=(RSPKLB-ISPKLB)*SPEK2(J,ISPKLB)+
     *   (1+ISPKLB-RSPKLB)*SPEK2(J,ISPKLB+1)
   10 CONTINUE
C
      RDATLB=((LABDAI-LMIN2)/STEP2)
      IDATLB=INT(RDATLB)
      DATAM(I)=(RDATLB-IDATLB)*DATA(IDATLB)+
     *   (1+IDATLB-RDATLB)*DATA(IDATLB+1)
C
   20 CONTINUE
C
C     LEAST SQUARES FIT OF THE EXPERIMENTAL SPECTRA
C     *********************************************
C
      DO 38 J=0,NFIT
      DO 36 I=0,NBER
      SPEK2(I,J)=SPEKM(I,J)
   36 CONTINUE
      DATAR(J)=DATAM(J)
   38 CONTINUE
      CALL FIT(NFIT+1,NBER,DATAR,SPEK2,UIT,RES)
C
C     Subroutine FIT determines a least squares fit of a function
C     DATA by a number of fit-functions stored in a matrix MATRIX. An
C     explanation of the heading of FIT is included in the source
C     text of FIT.
C
C     GRAPHICAL REPRESENTATION OF THE EXPERIMENTAL AND SIMULATED
C     EMISSION SPECTRA
C     **********************************************************
C
      DO 26 J=0,1019
      HOR(J)=LMIN2+STEP2*J
   26 CONTINUE
C
      DO 28 J=0,NFIT-1
      DATAM2(J)=0
      DO 27 I=0,NBER
      DATAM2(J)=DATAM2(J)+UIT(I)*SPEKM(I,J)
      HOR2(J)=LMINFI+J*STEPFI
   27 CONTINUE
   28 CONTINUE
C
      CALL POLYDZ(90,0,0,25,20,1,NDATA,HOR,DATA,10,LMINFI,LMAXFI,
     *    12,0,12000,.FALSE.,.FALSE.)
      CALL POLYDZ(90,0,0,25,20,1,NFIT,HOR2,DATAM2,10,LMINFI,LMAXFI,
```

```
     *    12,0,12000,.FALSE.,.FALSE.)
C
   29 CONTINUE
C
      CLOSE(6,DISP=CRUNCH)
      END
C
C
      SUBROUTINE FIT(NMEETP,NPAR,DATA,MATRIX,UIT,RES)
C
C     In this subroutine a function, which is known in NMEETP
C     points and stored in array DATA, is least squares fitted
C     to a set of NPAR fit-functions, stored in a matrix MATRIX.
C     The resultant contributions of all individual fit-functions
C     are stored in an array UIT. After the fit the variable RES
C     contains the normalized residu.
C
      REAL*4 BETA,RES,DD,DATA,UIT,MATRIX,SOM,DATARE,A
      INTEGER NMEETP,NPAR,INDEX,I,J,K,M
      DIMENSION DATA(0:1019),DD(0:180),UIT(0:180),
     *   MATRIX(0:50,0:1019),DATARE(0:1019)
C
      DO 150 K=0,NPAR-1
      SOM=0
C
      DO 110 I=K,NMEETP-1
      SOM=SOM+MATRIX(K,I)**2
  110 CONTINUE
C
      IF(MATRIX(K,K).GT.0) DD(K)=-SQRT(SOM)
      IF(MATRIX(K,K).LE.0) DD(K)=SQRT(SOM)
      MATRIX(K,K)=MATRIX(K,K)-DD(K)
      BETA=-DD(K)*MATRIX(K,K)
      IF(K.EQ.NPAR-1) GOTO 145
C
      DO 140 J=K+1,NPAR-1
      SOM=0
      DO 120 I=K,NMEETP-1
      SOM=SOM+MATRIX(K,I)*MATRIX(J,I)
  120 CONTINUE
      SOM=SOM/BETA
      MATRIX(J,I)=MATRIX(J,I)-SOM*MATRIX(K,I)
  130 CONTINUE
  140 CONTINUE
C
  145 CONTINUE
  150 CONTINUE
C
      DO 151 I=0,NMEETP-1
      DATARE(I)=DATA(I)
  151 CONTINUE
C
      DO 180 K=0,NPAR-1
      SOM=0
      DO 160 I=K,NMEETP-1
      SOM=SOM+MATRIX(K,I)*DATA(I)/(-DD(K)*MATRIX(K,K))
  160 CONTINUE
      DO 170 I=K,NMEETP-1
      DATA(I)=DATA(I)-SOM*MATRIX(K,I)
  170 CONTINUE
  180 CONTINUE
C
```

```
      UIT(NPAR-1)=DATA(NPAR-1)/DD(NPAR-1)
      DO 200 K=1,NPAR-1
      INDEX=NPAR-1-K
      UIT(INDEX)=DATA(INDEX)/DD(INDEX)
      DO 190 J=INDEX+1,NPAR-1
      UIT(INDEX)=UIT(INDEX)-MATRIX(J,INDEX)*UIT(J)/DD(INDEX)
  190 CONTINUE
  200 CONTINUE
C
      RES=0
      DO 210 I=NPAR,NMEETP-1
      RES=RES+DATA(I)**2
  210 CONTINUE
C
      WRITE(3,/) RES
      WRITE(6,/) VIB,NBER
      WRITE(6,/) UIT
  220 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE SNIJPU(F,I1,I2,E,R3)
C
C     For explanation see MINIB2/S/GEBONDENVRIJ/AANPASSINGBX
C
      REAL F,E,R3
      INTEGER I1,I2,I3
      DIMENSION F(0:4001)
C
      IF((I2-I1).LE.1) GOTO 30
      IF(F(I1).LT.E.AND.F(I2).LT.E) R3=-1
      IF(F(I1).GE.E.AND.F(I2).GE.E) R3=-1
      IF(R3.LT.0) GOTO 40
C
      IF(F(I2).GE.F(I1)) GOTO 20
   10 I3=INT((I2+I1)/2)
      IF(F(I3).LT.E) I2=I3
      IF(F(I3).GT.E) I1=I3
      IF((I2-I1).LE.1) GOTO 30
      GOTO 10
C
   20 I3=INT((I2+I1)/2)
      IF(F(I3).LT.E) I1=I3
      IF(F(I3).GE.E) I2=I3
      IF((I2-I1).LE.1) GOTO 30
      GOTO 20
C
   30 R3=I1+(F(I1)-E)/(F(I1)-F(I2))
   40 CONTINUE
C
      RETURN
      END
```

```
MINIB2/S/GEBONDENVRIJ/VIBFIT/TRIPEL
DATE & TIME PRINTED: MONDAY, JULY 20, 1987 @ 10:48:54.
```

```
C      MINIB2/S/GEBONDENVRIJ/VIBFIT/TRIPEL
C
C      This program is part of a series of programs to simulate
C      bound-free rare gas halide emission spectra.
C      In this program an experimental emission spectrum from two
C      or three bound-free electronic transitions, stored on FILE 2,
C      is simulated using calculated emission spectra from
C      individual emission spectra (or collections of emission
C      spectra from individual vibrational states) stored on
C      specified FILES 1, 8 (and 7). Using a least squares method
C      vibrational distributions of the upper states are determined,
C      and stored on a file MINIB2/S/GEBONDENVRIJ/FITUITVOER (FILE 6).
C      The program also creates a previewfile which contains a
C      graphical representation of the experimental emission spectrum
C      and the simulated spectrum, also divided into contributions
C      from separate electronic transitions. It should be noted that
C      in the present form, it is assumed that the calculated
C      spectra on FILES 1 and 7 correspond to the same upper state,
C      this program being written for a simultaneous treatment of
C      B-X(1/2) (FILE 1), C-A(3/2) (FILE 8) and B-A(1/2) spectra
C      (FILE 7).
C
C      *****************************************************************
C
C      DECLARATIONS
C      ************
C
FILE  1(KIND=DISK,FILETYPE=7,TITLE="MINIB2/D/GEBONDENVRIJ/XEBRBX/5")
FILE  2(KIND=DISK,FILETYPE=7,TITLE="MINIB2/D/GEBONDENVRIJ/DATAOUT")
FILE  3(KIND=REMOTE)
FILE  4(KIND=REMOTE)
FILE  5(KIND=PRINTER)
FILE  6(KIND=DISK,TITLE="MINIB2/D/GEBONDENVRIJ/FITUITVOER",
     *  PROTECTION=SAVE,NEWFILE=TRUE)
C  FILE  7(KIND=DISK,FILETYPE=7,TITLE="MINIB2/D/GEBONDENVRIJ/XEBRBA/2")
FILE  8(KIND=DISK,FILETYPE=7,TITLE="MINIB2/D/GEBONDENVRIJ/XEBRCA/5")
      BLOCK GLOBALS
 $ INCLUDE "PLOTTER/FORTRAN/DECLARATION ON APPL"
      END
 $ INCLUDE "PLOTTER/FORTRAN/POLYDZ ON APPL"
 $ INCLUDE "PLOTTER/FORTRAN/MPOLYD ON APPL"
      REAL*4 STEP3,UMINUP,EU,SPEK,STEP2,DATA,STEPFI,LABDAI,
     *   STEP3C,UMINUC,EUC,SPEKC,RSPKLC,SPEKMC,SPEK2C,
     *   STEP3B,UMINUB,SPEKB,SPEK2B,HULP1,HULP2,FV,FITF,MAT,
     *   RSPKLB,RDATLB,SPEKM,DATAM,UIT,RES,SPEKMA,DATAMA,
     *   CORR,HOR,HOR2,DATAM2,SPEK2,LAB,DATAR,SP,DATAB,DATAC,GEM,
     *   LMIN2,LMAX2,LMINC,LMAXC
      INTEGER*4 LMIN,LMAX,VIB,VIBF,STEP,LMINFI,LMAXFI,
     *   NFIT,NDATA,I,J,ISPKLB,IDATLB,IKAN,NKAN,SBER,NBER,
     *   VIBC,VIBFC,ISPKLC,SBERC,NBERC,LMINB,LMAXB
      DIMENSION EU(0:179),SPEK(0:90,0:1019),SPEKM(0:50,0:1019),
     *   DATA(0:1019),DATAM2(0:1019),DATAM(0:1019),UIT(0:180),
     *   HOR(0:1019),SPEKC(0:90,0:1019),SPEK2C(0:20,0:1019),
     *   SPEKB(0:90,0:1019),SPEK2B(0:20,0:1019),
     *   HOR2(0:1019),SPEK2(0:50,0:1019),EUC(0:179),
     *   LAB(0:1019),DATAR(0:1019),SP(0:20000),DATAB(0:1019),
     *   DATAC(0:1019),HULP1(0:1019),HULP2(0:1019),RESIDU(0:1019),
     *   FV(0:20),FITF(0:20),MAT(0:1,0:20)
```

```
C
C     INPUT VARIABLES AND INITIALIZATION
C     **********************************
C
      READ(1,/) LMIN,LMAX,STEP3
      READ(1,/) VIBF,STEP
      READ(1,/) UMINUP
      READ(1,/) EU
C
      READ(8,/) LMINC,LMAXC,STEP3C
      READ(8,/) VIBFC,STEPC
      READ(8,/) UMINUC
      READ(8,/) EUC
C
C     READ(7,/) LMINB,LMAXB,STEP3B
C     READ(7,/) VIBF,STEP
C     READ(7,/) UMINUB
C     READ(7,/) EU
C
      WRITE(3,1) VIBF,STEP
  1   FORMAT(´ VIBR. NIVEAUS B-X BEREKEND TOT V=´,I4,/,
     *   ´ STAPGROOTTE=´,I3,/)
C 1   FORMAT(´ VIBR. STATES B-X DETERMINED UP TO V=´,I4,/,
C    *   ´ INTERVALSIZE=´,I3,/)
C
      WRITE(3,3) VIBFC,STEPC
  3   FORMAT(´ VIBR. NIVEAUS C-A BEREKEND TOT V=´,I4,/,
     *   ´ STAPGROOTTE=´,I3,/)
C 3   FORMAT(´ VIBR. STATES C-A DETERMINED UP TO V=´,I4,/,
C    *   ´ INTERVALSIZE=´,I3,/)
C
      WRITE(3,4)
  4   FORMAT(´ GEEF VBER EN SBER, RESP. VBERC EN SBERC:´)
C 4   FORMAT(´ ENTER VBER AND SBER, AND VBERC AND SBERC, RESPECTIVELY:´)
  2   READ(4,/) VIB, SBER, VIBC, SBERC
C
      NBER=INT(VIB/SBER)
      NBERC=INT(VIBC/SBERC)
      WRITE(6,/) VIB,SBER
      WRITE(6,/)VIBC,SBERC
C
      DO 40 I=0,VIB-1
      DO 30 J=1,34
  30  READ(1) (HULP1(30*(J-1)+K),K=0,29)
      DO 133 J=0,1019
  133 SPEK(I,J)=HULP1(J)
  40  CONTINUE
C
      DO 140 I=0,VIBC-1
      DO 130 J=1,34
  130 READ(8) (HULP1(30*(J-1)+K),K=0,29)
      DO 132 J=0,1019
  132 SPEKC(I,J)=HULP1(J)
  140 CONTINUE
C
C     DO 240 I=0,VIB-1
C     DO 230 J=1,34
C 230 READ(7) (HULP1(30*(J-1)+K),K=0,29)
C     DO 232 J=0,1019
C 232 SPEKB(I,J)=HULP1(J)
C 240 CONTINUE
C
```

```
      READ(2,/) LMIN2,LMAX2,STEP2
      READ(2,/) DATA
      NDATA=INT((LMAX2-LMIN2)/STEP2)
C
      DO 400 I=1,5
      GEM=GEM+DATA(I)/10.
  400 CONTINUE
C
      DO 401 I=NDATA-5,NDATA-1
      GEM=GEM+DATA(I)/10.
  401 CONTINUE
C
      DO 402 I=1,NDATA-1
      DATA(I)=DATA(I)-GEM
  402 CONTINUE
C
      CORR=-11.9
      LMIN2=LMIN2+CORR
      LMAX2=LMAX2+CORR
C
C     NORMALIZATION OF CALCULATED AND EXPERIMENTAL EMISSION SPECTRA
C     ************************************************************
C
      SPEKMA=0
      DO 61 I=0,VIB-1
      DO 60 J=0,1019
      IF(SPEK(I,J).GT.SPEKMA) SPEKMA=SPEK(I,J)
   60 CONTINUE
   61 CONTINUE
C
      DO 63 I=0,VIB-1
      DO 62 J=0,1019
      SPEK(I,J)=SPEK(I,J)/SPEKMA
   62 CONTINUE
   63 CONTINUE
C
      SPEKMC=0
      DO 69 I=0,VIBC-1
      DO 68 J=0,1019
      IF(SPEKC(I,J).GT.SPEKMC) SPEKMC=SPEKC(I,J)
   68 CONTINUE
   69 CONTINUE
C
      DO 71 I=0,VIBC-1
      DO 70 J=0,1019
      SPEKC(I,J)=SPEKC(I,J)/SPEKMC
   70 CONTINUE
   71 CONTINUE
C
      SPEKMB=0
      DO 89 I=0,VIB-1
      DO 88 J=0,1019
      IF(SPEKB(I,J).GT.SPEKMB) SPEKMB=SPEKB(I,J)
   88 CONTINUE
   89 CONTINUE
C
      DO 87 I=0,VIB-1
      DO 86 J=0,1019
      SPEKB(I,J)=SPEKB(I,J)/SPEKMA
   86 CONTINUE
   87 CONTINUE
C     WRITE(3,/) SPEKMA,SPEKMC,SPEKMB
```

```
C
      DATAMA=0
      DO 64 J=0,1019
      IF(DATA(J).GT.DATAMA) DATAMA=DATA(J)
   64 CONTINUE
C
      DO 65 J=0,1019
      DATA(J)=DATA(J)/DATAMA*10000
   65 CONTINUE
   66 CONTINUE
C
C     DETERMINATION OF WAVELENGHT INTERVAL AND STEPSIZE FOR LEAST
C     SQUARES FIT
C     **********************************************************
C
      LMINFI=LMIN
      LMAXFI=LMAXC
      IF(LMIN2.GT.LMINFI) LMINFI=LMIN2
      IF(LMAX2.LT.LMAXFI) LMAXFI=LMAX2
C
      STEPFI=STEP3
      IF(STEP3C.GT.STEPFI) STEPFI=STEP3C
      IF(STEP2.GT.STEPFI) STEPFI=STEP2
      NFIT=INT((LMAXFI-LMINFI)/STEPFI)
C
C     CREATION OF FIT-FUNCTIONS FOR LEAST SQUARES FIT
C     **********************************************
C
C     a) Joining of SBER/SBERC emission spectra of individual
C        vibrational states into a single fit-function.
C
      IF(SBER.EQ.1) GOTO 17
      DO 17 I=0,1019
      DO 15 J=0,NBER-1
      SPEK(J,I)=SPEK(SBER*J,I)
      DO 16 K=1,SBER-1
      SPEK(J,I)=SPEK(J,I)+SPEK(SBER*J+K,I)
   16 CONTINUE
      SPEK(J,I)=SPEK(J,I)/SBER
   15 CONTINUE
   17 CONTINUE
C
      IF(SBERC.EQ.1) GOTO 77
      DO 77 I=0,1019
      DO 75 J=0,NBERC-1
      SPEKC(J,I)=SPEKC(SBERC*J,I)
      DO 76 K=1,SBERC-1
      SPEKC(J,I)=SPEKC(J,I)+SPEKC(SBERC*J+K,I)
   76 CONTINUE
      SPEKC(J,I)=SPEKC(J,I)/SBERC
   75 CONTINUE
   77 CONTINUE
C
      IF(SBER.EQ.1) GOTO 97
      DO 97 I=0,1019
      DO 95 J=0,NBER-1
      SPEKB(J,I)=SPEKB(SBER*J,I)
      DO 96 K=1,SBER-1
      SPEKB(J,I)=SPEKB(J,I)+SPEKB(SBER*J+K,I)
   96 CONTINUE
      SPEKB(J,I)=SPEKB(J,I)/SBER
   95 CONTINUE
```

```
   97 CONTINUE
C
C     b) The introduction of a spectral resolution into the fit-
C        functions
C
      NKAN=3
      DO 35 IKAN=-NKAN,NKAN
C
      DO 34 J=0,NBER-1
      DO 33 I=0,1019
      IF(I+IKAN.GE.0.AND.I+IKAN.LE.1019) GOTO 41
      SPEK2(J,I)=SPEK2(J,I)+SPEK(J,I)
      GOTO 32
   41 CONTINUE
      SPEK2(J,I)=SPEK2(J,I)+SPEK(J,I+IKAN)
   32 CONTINUE
   33 CONTINUE
   34 CONTINUE
C
      DO 345 J=0,NBERC-1
      DO 344 I=0,1019
      IF(I+IKAN.GE.0.AND.I+IKAN.LE.1019) GOTO 342
      SPEK2C(J,I)=SPEK2C(J,I)+SPEKC(J,I)
      GOTO 343
  342 CONTINUE
      SPEK2C(J,I)=SPEK2C(J,I)+SPEKC(J,I+IKAN)
  343 CONTINUE
  344 CONTINUE
  345 CONTINUE
C
      DO 434 J=0,NBER-1
      DO 433 I=0,1019
      IF(I+IKAN.GE.0.AND.I+IKAN.LE.1019) GOTO 441
      SPEK2B(J,I)=SPEK2B(J,I)+SPEKB(J,I)
      GOTO 432
  441 CONTINUE
      SPEK2B(J,I)=SPEK2B(J,I)+SPEKB(J,I+IKAN)
  432 CONTINUE
  433 CONTINUE
  434 CONTINUE
C
   35 CONTINUE
C
C     CONVERSION OF CALCULATED AND EXPERIMENTAL EMISSION SPECTRA
C     TO SPECTRA MATCHING THE REQUIRED WAVELENGTH INTERVAL AND
C     HAVING THE CORRECT STEPSIZE
C     **********************************************************
C
      DO 20 I=0,NFIT
      LABDAI=LMINFI+I*STEPFI
C
      IF(LABDAI.GT.LMAX) GOTO 11
      RSPKLB=((LABDAI-LMIN)/STEP3)
      ISPKLB=INT(RSPKLB)
      DO 10 J=0,NBER-1
      SPEKM(J,I)=(RSPKLB-ISPKLB)*SPEK2(J,ISPKLB)+
     *   (1+ISPKLB-RSPKLB)*SPEK2(J,ISPKLB+1)
   10 CONTINUE
   11 CONTINUE
C
      IF(LABDAI.LT.LMINC) GOTO 12
      RSPKLC=((LABDAI-LMINC)/STEP3C)
```

```
      ISPKLC=INT(RSPKLC)
      DO 13 J=0,NBERC-1
      SPEKM(J+NBER,I)=(RSPKLC-ISPKLC)*SPEK2C(J,ISPKLC)+
     *   (1+ISPKLC-RSPKLC)*SPEK2C(J,ISPKLC+1)
   13 CONTINUE
   12 CONTINUE
C
      SPEKM(NBER+NBERC,I)=1
C
C     IF(LABDAI.LT.LMINB) GOTO 14
C     RSPKLB=((LABDAI-LMINB)/STEP3B)
C     ISPKLB=INT(RSPKLB)
C     DO 18 J=0,NBER-1
C     SPEKM(J,I)=SPEKM(J,I)+(RSPKLB-ISPKLB)*SPEK2B(J,ISPKLB)+
C    *   (1+ISPKLB-RSPKLB)*SPEK2B(J,ISPKLB+1)
C  18 CONTINUE
C  14 CONTINUE
C
      RDATLB=((LABDAI-LMIN2)/STEP2)
      IDATLB=INT(RDATLB)
      IF(I.LE.0) GOTO 20
      DATAM(I)=(RDATLB-IDATLB)*DATA(IDATLB)+
     *   (1+IDATLB-RDATLB)*DATA(IDATLB+1)
   20 CONTINUE
C
C     LEAST SQUARES FIT OF THE EXPERIMENTAL SPECTRUM
C     *********************************************
C
      DO 38 J=0,NFIT
      DO 36 I=0,NBER+NBERC-1
      SPEK2(I,J)=SPEKM(I,J)
   36 CONTINUE
      DATAR(J)=DATAM(J)
   38 CONTINUE
      CALL FIT(NFIT+1,NBER+NBERC,DATAR,SPEK2,UIT,RES)
C
C     Subroutine FIT determines a least squares fit of a function
C     DATA by a number of fit-functions in stored in a matrix MATRIX.
C     An explanation of the heading of FIT is included in the source
C     text of FIT.
C
C     GRAPHICAL REPRESENTATION OF THE EXPERIMENTAL AND SIMULATED
C     EMISSION SPECTRA
C     *********************************************************
C
      DO 26 J=0,1019
      HOR(J)=LMIN2+STEP2*J
   26 CONTINUE
C
      DO 28 J=0,NFIT-1
      DATAB(J)=0
      DATAC(J)=0
      DATAM2(J)=0
C
      DO 27 I=0,NBER-1
C     IF(UIT(I).LT.0) GOTO 27
      DATAB(J)=DATAB(J)+UIT(I)*SPEKM(I,J)
   27 CONTINUE
C
      DO 39 I=NBER,NBER+NBERC-1
C     IF(UIT(I).LT.0) GOTO 39
      DATAC(J)=DATAC(J)+UIT(I)*SPEKM(I,J)
```

```
   39 CONTINUE
C
      HOR2(J)=LMINFI+J*STEPFI
      DATAB(J)=DATAB(J)+UIT(NBER+NBERC)
      DATAM2(J)=DATAB(J)+DATAC(J)
      DATAC(J)=DATAC(J)+UIT(NBER+NBERC)
      RESIDU(J)=DATAM(J)-DATAM2(J)+11000
C
   28 CONTINUE
C     GOTO 29
      CALL POLYDZ(90,0,0,25,20,1,NFIT,HOR2,DATAM,10,LMIN,LMAXC,
     *     12,0,12000,.FALSE.,.FALSE.)
      CALL POLYDZ(90,0,0,25,20,1,NFIT,HOR2,DATAM2,10,LMIN,LMAXC,
     *     12,0,12000,.FALSE.,.FALSE.)
      CALL POLYDZ(90,0,0,25,20,1,NFIT,HOR2,DATAB,10,LMIN,LMAXC,
     *     12,0,12000,.FALSE.,.FALSE.)
      CALL POLYDZ(90,0,0,25,20,1,NFIT,HOR2,DATAC,10,LMIN,LMAXC,
     *     12,0,12000,.FALSE.,.FALSE.)
      CALL POLYDZ(90,0,0,25,20,1,NFIT,HOR2,RESIDU,10,LMIN,LMAXC,
     *     12,0,12000,.FALSE.,.FALSE.)
C
C     DETERMINATION OF THE SURPRISAL PARAMETERS, IF THE VIBRATIONAL
C     DISTRIBUTIONS DETERMINED ARE SUPPOSED TO BE LINEAR SURPRISAL
C     DISTRIBUTIONS
C     ************************************************************
C
C     Note: this calculation can only be successful if all UIT(I)
C     are > 0.
C
      EBIND=52.7
      EMAX=8065*8.315-349.83*EBIND
C
      DO 410 I=0,NBER-1
      DO 411 J=0,SBER-1
      FV(I)=FV(I)+(EU(I*SBER+J)-UMINUP)/(EMAX-UMINUP)/SBER
  411 CONTINUE
C
      IF(FV(I).GE.1) GOTO 410
      FITF(I)=ALOG(ABS(UIT(I)))-1.5*ALOG(1-FV(I))
      SPEK2(0,I)=FV(I)
      SPEK2(1,I)=1.
      NMEET=I+1
  410 CONTINUE
C
      CALL FIT(NMEET,2,FITF,SPEK2,UIT,RES)
      WRITE(3,/) UIT(0),UIT(1)
C
      DO 412 I=0,NBERC-1
      DO 413 J=0,SBERC-1
      FV(I)=FV(I)+(EUC(I*SBERC+J)-UMINUC)/(EMAX-UMINUC)/SBERC
  413 CONTINUE
C
      IF(FV(I).GE.1) GOTO 412
      FITF(I)=ALOG(ABS(UIT(I+NBER)))-1.5*ALOG(1-FV(I))
      SPEK2(0,I)=FV(I)
      SPEK2(1,I)=1.
      NMEET=I+1
  412 CONTINUE
C
      CALL FIT(NMEET,2,FITF,SPEK2,UIT,RES)
      WRITE(3,/) UIT(0),UIT(1)
C
```

```
   29 CONTINUE
C
      CLOSE(6,DISP=CRUNCH)
      END
C
      SUBROUTINE FIT(NMEETP,NPAR,DATA,MATRIX,UIT,RES)
C
C     In this subroutine a function, which is known in NMEETP
C     points and stored in array DATA, is least squares fitted
C     to a set of NPAR fit-functions, stored in a matrix MATRIX.
C     The resultant contributions of all individual fit-functions
C     are stored in an array UIT. After the fit the variable RES
C     contains the normalized residu.
C
      REAL*4 BETA,RES,DD,DATA,UIT,MATRIX,SOM,DATARE,A
      INTEGER NMEETP,NPAR,INDEX,I,J,K,M
      DIMENSION DATA(0:1019),DD(0:180),UIT(0:180),
     *   MATRIX(0:50,0:1019),DATARE(0:1019)
C
      DO 150 K=0,NPAR-1
      SOM=0
C
      DO 110 I=K,NMEETP-1
      SOM=SOM+MATRIX(K,I)**2
  110 CONTINUE
C
      IF(MATRIX(K,K).GT.0) DD(K)=-SQRT(SOM)
      IF(MATRIX(K,K).LE.0) DD(K)=SQRT(SOM)
      MATRIX(K,K)=MATRIX(K,K)-DD(K)
      BETA=-DD(K)*MATRIX(K,K)
      IF(K.EQ.NPAR-1) GOTO 145
C
      DO 140 J=K+1,NPAR-1
      SOM=0
      DO 120 I=K,NMEETP-1
      SOM=SOM+MATRIX(K,I)*MATRIX(J,I)
  120 CONTINUE
      SOM=SOM/BETA
      DO 130 I=K,NMEETP-1
      MATRIX(J,I)=MATRIX(J,I)-SOM*MATRIX(K,I)
  130 CONTINUE
  140 CONTINUE
C
  145 CONTINUE
  150 CONTINUE
C
      DO 151 I=0,NMEETP-1
      DATARE(I)=DATA(I)
  151 CONTINUE
C
      DO 180 K=0,NPAR-1
      SOM=0
      DO 160 I=K,NMEETP-1
      SOM=SOM+MATRIX(K,I)*DATA(I)/(-DD(K)*MATRIX(K,K))
  160 CONTINUE
      DO 170 I=K,NMEETP-1
      DATA(I)=DATA(I)-SOM*MATRIX(K,I)
  170 CONTINUE
  180 CONTINUE
C
      UIT(NPAR-1)=DATA(NPAR-1)/DD(NPAR-1)
      DO 200 K=1,NPAR-1
```

```
      INDEX=NPAR-1-K
      UIT(INDEX)=DATA(INDEX)/DD(INDEX)
      DO 190 J=INDEX+1,NPAR-1
      UIT(INDEX)=UIT(INDEX)-MATRIX(J,INDEX)*UIT(J)/DD(INDEX)
  190 CONTINUE
  200 CONTINUE
C
      RES=0
      DO 210 I=NPAR,NMEETP-1
      RES=RES+DATA(I)**2
  210 CONTINUE
C
      WRITE(3,/) RES
C     WRITE(3,/) UIT
      WRITE(6,/) UIT
  220 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE SNIJPU(F,I1,I2,E,R3)
C
C     For an explanation see MINIB2/S/GEBONDENVRIJ/AANPASSINGBX
C
      REAL F,E,R3
      INTEGER I1,I2,I3
      DIMENSION F(0:4001)
C
      IF((I2-I1).LE.1) GOTO 30
      IF(F(I1).LT.E.AND.F(I2).LT.E) R3=-1
      IF(F(I1).GE.E.AND.F(I2).GE.E) R3=-1
      IF(R3.LT.0) GOTO 40
C
      IF(F(I2).GE.F(I1)) GOTO 20
   10 I3=INT((I2+I1)/2)
      IF(F(I3).LT.E) I2=I3
      IF(F(I3).GT.E) I1=I3
      IF((I2-I1).LE.1) GOTO 30
      GOTO 10
C
   20 I3=INT((I2+I1)/2)
      IF(F(I3).LT.E) I1=I3
      IF(F(I3).GE.E) I2=I3
      IF((I2-I1).LE.1) GOTO 30
      GOTO 20
C
   30 R3=I1+(F(I1)-E)/(F(I1)-F(I2))
   40 CONTINUE

      RETURN
      END
```

MINIB2/S/GEBONDENVRIJ/VIBPLOTTER
DATE & TIME PRINTED: MONDAY, JULY 20, 1987 @ 10:49:17.

```
C     MINIB2/S/GEBONDENVRIJ/VIBPLOTTER
C
C     This program is part of a series of programs to simulate
C     bound-free rare gas halide emission spectra.
C     With this program a graphical representation can be given of
C     a vibrational distribution determined in MINIB2/S/GEBONDENVRIJ/
C     VIBFIT.
C
C***********************************************************************
C
FILE  1(KIND=DISK,FILETYPE=7,TITLE="MINIB2/D/GEBONDENVRIJ/FITUITVOER")
      BLOCK GLOBALS
 $ INCLUDE "PLOTTER/FORTRAN/DECLARATION ON APPL"
      END
 $ INCLUDE "PLOTTER/FORTRAN/POLYDZ ON APPL"
      REAL UIT,UIT2,UITMAX,HOR
      INTEGER M,INC,VIB,N
      DIMENSION UIT(0:180),HOR(0:180),UIT2(0:180)
C
      READ(1,/) VIB,INC
      N=VIB/INC-1
C
      UITMAX=0
      READ(1,/) UIT
      DO 10 I=0,N
      IF(UIT(I).GT.UITMAX) UITMAX=UIT(I)
   10 CONTINUE
C
      DO 30 I=0,N
      HOR(2*I)=I*INC
      HOR(2*I+1)=(I+1)*INC-.01
      UIT2(2*I)=UIT(I)/UITMAX
      UIT2(2*I+1)=UIT(I)/UITMAX
   30 CONTINUE
      HOR(2*N+2)=(N+1)*INC
      UIT2(2*N+2)=0
C
      CALL POLYDZ(90,0,0,25,20,1,2*N+3,HOR,UIT2,10,0,VIB,7,-.2,1.2,
     *   .FALSE.,.FALSE.)
      STOP
      END
```

MINIB2/S/GEBONDENVRIJ/VIBPLOTTER/DUBBEL
DATE & TIME PRINTED: THURSDAY, JULY 30, 1987 @ 14:32:41.

```
C      MINIB2/S/GEBONDENVRIJ/VIBPLOTTER/DUBBEL
C
C      This program is part of a series of programs to simulate
C      bound-free rare gas halide emission spectra.
C      With this program a graphical representation can be given of
C      the vibrational distributions determined in MINIB2/S/
C      GEBONDENVRIJ/VIBFIT/TRIPEL.
C
C********************************************************************
C
FILE  1(KIND=DISK,FILETYPE=7,TITLE="MINIB2/D/GEBONDENVRIJ/FITUITVOER")
      BLOCK GLOBALS
 $ INCLUDE "PLOTTER/FORTRAN/DECLARATION ON APPL"
      END
 $ INCLUDE "PLOTTER/FORTRAN/POLYDZ ON APPL"
      REAL UIT,UIT2,UIT3,UITMA2,UITMAX,HOR,HOR2
      INTEGER M,INC,INC2,VIB,VIB2,N,N2
      DIMENSION UIT(0:180),HOR(0:180),UIT2(0:180),UIT3(0:180),
     *   HOR2(0:180)
C
      READ(1,/) VIB,INC
      READ(1,/) VIB2,INC2
      VIB=10*VIB
      INC=10*INC
      VIB2=15*VIB2
      INC2=15*INC2
      N=VIB/INC-1
      N2=VIB2/INC2-1

      UITMAX=0
      UITMA2=0
      READ(1,/) UIT
      DO 10 I=0,N
      IF(UIT(I).GT.UITMAX) UITMAX=UIT(I)
   10 CONTINUE
      DO 11 I=0,N2
      IF(UIT(I+N+1).GT.UITMA2) UITMA2=UIT(I+N+1)
   11 CONTINUE
C
      DO 30 I=0,N
      HOR(2*I)=I*INC
      HOR(2*I+1)=(I+1)*INC-.01
      UIT2(2*I)=UIT(I)/UITMAX
      UIT2(2*I+1)=UIT(I)/UITMAX
   30 CONTINUE
C
      DO 31 I=0,N2
      HOR2(2*I)=I*INC2
      HOR2(2*I+1)=(I+1)*INC2-.01
      UIT3(2*I)=UIT(I+N+1)/UITMA2+1.5
      UIT3(2*I+1)=UIT(I+N+1)/UITMA2+1.5
   31 CONTINUE
C
      HOR(2*N+2)=(N+1)*INC
      UIT2(2*N+2)=0
      HOR2(2*N2+2)=(N2+1)*INC2
      UIT3(2*N2+2)=1.5
C
```

```
      NVIB=VIB
      IF(VIB2.GE.NVIB) NVIB=VIB2
      CALL POLYDZ(90,0,0,25,20,1,2*N+3,HOR,UIT2,10,0,NVIB,7,-.5,3.0,
     *   .FALSE.,.FALSE.)
      CALL POLYDZ(90,0,0,25,20,1,2*N2+3,HOR2,UIT3,10,0,NVIB,7,-.5,
     *   3.0,.FALSE.,.FALSE.)

      STOP
      END
```

MINIB2/S/GEBONDENVRIJ/VIBCALC
DATE & TIME PRINTED: MONDAY, JULY 20, 1987 @ 10:49:51.

```
C     MINIB2/S/GEBONDENVRIJ/VIBCALC
C
C     This program is part of a series of programs to simulate
C     bound-free rare gas halide emission spectra.
C     In this program a bound-free emission spectrum is calculated,
C     assuming a vibrational distribution specified either on a FILE 3
C     or on the terminal, and using calculated emission spectra of
C     individual vibrational states stored on a specified FILE 4.
C
C******************************************************************
C
C     DECLARATIONS
C     ************
C
FILE  1(KIND=REMOTE)                  % WRITE TO TERMINAL
FILE  2(KIND=REMOTE)                  % READ FROM TERMINAL
FILE  3(KIND=DISK,FILETYPE=7,TITLE="MINIB2/D/GEBONDENVRIJ/VIBINVOER")
FILE  4(KIND=DISK,TITLE="MINIB2/D/GEBONDENVRIJ/XEBRBX/5")
FILE  5(KIND=DISK,TITLE="MINIB2/D/GEBONDENVRIJ/VIBUITVOER",
     *   PROTECTION=SAVE,NEWFILE=TRUE)
FILE  6(KIND=PRINTER)
      REAL*4 STEP3,UMINUP,EU,SPEK,LV,DUMMY,C1,TBOLTZ,EBIND,
     *   EMETA,KBOLTZ,SPUIT
      INTEGER*4 SQUEST,INTACT,LMIN,LMAX,VIB,STEP,KEUS,I,J,KMAX
      DIMENSION SPEK(1:180,0:1019),EU(180),P(180),SPUIT(0:1019)
C
C     INPUT VARIABLES
C     ***************
C
      SQUEST=0
      INTACT=3
      WRITE(1,1)
      READ(2,/) SQUEST
  1   FORMAT(´ INVOER INTERACTIEF VANAF TERMINAL (JA=1)?´)
C 1   FORMAT(´ INPUT FROM TERMINAL (YES=1)?´)
      IF(SQUEST.EQ.1) INTACT=2
C
      SQUEST=0
      TERM=0
      IF(INTACT.EQ.2) WRITE(1,2)
      READ(INTACT,/) SQUEST
  2   FORMAT(´ AANMAAK PRINTERFILE MET PROGRAMMAGEGEVENS (JA=1)?´,/)
C 2   FORMAT(´ CREATE A PRINTERFILE WITH PROGRAM-INFO (YES=1)?´,/)
      IF(SQUEST.EQ.1) TERM=1
C
      READ(4,/) LMIN,LMAX,STEP3
      READ(4,/) VIB,STEP
      READ(4,/) UMINUP
      READ(4,/) EU
C
      DO 161 I=1,VIB
      DO 151 J=1,34
      READ(4) (SPEK(I,30*(J-1)+K),K=0,29)
  151 CONTINUE
  161 CONTINUE
C
C     SELECTION OF VIBRATIONAL DISTRIBUTION
C     *************************************
```

```
C
     IF(INTACT.EQ.2) WRITE(1,3) LMIN,LMAX,STEP3,VIB,STEP
     IF(TERM.EQ.1) WRITE(6,3) LMIN,LMAX,STEP3,VIB,STEP
  3  FORMAT(´ OP DE UITVOERFILE VAN PROGRAMMA MINIB2/S/GEBONDENVRIJ/´,
    *   /,´ POTCALC STAAN BEREKENDE EMISSIE-SPEKTRA IN EEN INTERVAL´,
    *   /,´ VAN ´,I3,´ TOT ´,I3,´ NM (STAPGROOTTE ´,F6.2,´ NM), VOOR´,
    *   /,I4,´ VIBRATIONELE NIVEAUS (STAPGROOTTE VIBRATIONEEL´,
    *   /,´ QUANTUMGETAL= ´,I2,´).´,/)
C 3  FORMAT(´ ON THE OUTPUTFILE OF PROGRAM MINIB2/S/GEBONDENVRIJ/´,
C   *   /,´ POTCALC CALCULATED EMISSION SPECTRA ARE STORED IN A´,
C   *   /,´ WAVELENGTHINTERVAL BETWEEN ´,I3,´ AND ´,I3,´ NM ´,
C   *   /´, (STEPSIZE ´,F6.2,´ NM), FOR ´,I4,´ VIBRATIONAL STATES ´,
C   *   /´, (STEPSIZE VIBRATIONAL QUANTUMNUMBER= ´,I2,´).´,/)
C
     IF(INTACT.EQ.2) WRITE(1,4)
  4  FORMAT(´ KIES EEN VIBRATIONELE VERDELING (1-4):´,/,
    *   ´(NB. FV=EVIB/EVIBMX)´,/)
C 4  FORMAT(´ CHOOSE A VIBRATIONAL DISTRIBUTION (1-4):´,/,
C   *   ´(NOTE. FV=EVIB/EVIBMX)´,/)
C
     IF(INTACT.EQ.2) WRITE(1,5)
     IF(TERM.EQ.1) WRITE(6,5)
  5  FORMAT(´ 1)   P(FV)=((1-FV)**3/2)*EXP(-LV*FV)´,/,
    *   ´ 2)   P(FV)=C´,/,
    *   ´ 3)   P(FV)=(1+C1*FV)´,/,
    *   ´ 4)   P(FV)=EXP(-EVIB/EBOLTZ)´,/)
     READ(INTACT,/) KEUS
     IF(TERM.EQ.1) WRITE(6,6) KEUS
  6  FORMAT(´ GEMAAKTE KEUS= ´,I2)
C 6  FORMAT(´ YOUR CHOICE= ´,I2)
C
     IF(INTACT.EQ.2.AND.KEUS.EQ.1) WRITE(1,7)
  7  FORMAT(´ INVOER SURPRISAL PARAMETER LV= ´,/)
C 7  FORMAT(´ INPUT SURPRISAL PARAMETER LV= ´,/)
     IF(KEUS.EQ.1) READ(INTACT,/) LV
C
     IF(INTACT.EQ.3.AND.KEUS.EQ.2) READ(INTACT,/) DUMMY
C
     IF(INTACT.EQ.2.AND.KEUS.EQ.3) WRITE(1,8)
  8  FORMAT(´ INVOER CONSTANTE C1= ´)
C 8  FORMAT(´ INPUT CONSTANT C1= ´)
     IF(KEUS.EQ.3) READ(INTACT,/) C1
C
     IF(INTACT.EQ.2.AND.KEUS.EQ.4) WRITE(1,9)
  9  FORMAT(´ INVOER BOLTZMANNTEMPERATUUR= ´)
C 9  FORMAT(´ INPUT BOLTZMANNTEMPERATURE= ´)
     IF(KEUS.EQ.4) READ(INTACT,/) TBOLTZ
C
     IF(TERM.EQ.1.AND.KEUS.EQ.1) WRITE(6,10) LV
     IF(TERM.EQ.1.AND.KEUS.EQ.3) WRITE(6,11) C1
     IF(TERM.EQ.1.AND.KEUS.EQ.4) WRITE(6,12) TBOLTZ
  10 FORMAT(´ SURPRISAL PARAMETER LV= ´,F6.2,/)
  11 FORMAT(´ CONSTANTE C1= ´,F6.2,/)
  12 FORMAT(´ BOLTZMANNTEMPERATUUR= ´,F7.1,´ KELVIN´,/)
C
     IF(INTACT.EQ.2) WRITE(1,13)
     READ(INTACT,/) EBIND
     IF(INTACT.EQ.2) WRITE(1,14)
     READ(INTACT,/) EMETA
     EBIND=EBIND*349.83             % CONVERSION TO,CM-1
     EMETA=EMETA*349.83*23.06       % CONVERSION TO CM-1
  13 FORMAT(´ INVOER BINDINGSENERGIE (KCAL/MOL)´,/)
```

```
C 13  FORMAT(' INPUT BONDING ENERGY (KCAL/MOL)',/)
  14  FORMAT(' INVOER METASTABIELENENERGIE (EV)',/)
C 14  FORMAT(' INPUT METASTABLES ENERGY (EV)',/)
      KBOLTZ=1.38E-23
C
C     CALCULATION OF THE VIBRATIONAL DISTRIBUTION
C     *******************************************
C
      DO 100 I=1,VIB
      FV=(EU(I)-UMINUP)/(EMETA-EBIND-UMINUP)
      IF(FV.GE.1.AND.KEUS.EQ.1) GOTO 100
      IF(FV.GE.1.AND.KEUS.EQ.3) GOTO 100
      IF(KEUS.EQ.1) P(I)=(SQRT((1-FV)**3))*EXP(-LV*FV)
      IF(KEUS.EQ.2) P(I)=1.
      IF(KEUS.EQ.3) P(I)=1+C1*FV
      IF(KEUS.EQ.4) P(I)=EXP(-(EU(I)-UMINUP)/(KBOLTZ*TBOLTZ))
C     WRITE(1,/) I,FV,P(I)
  100 CONTINUE
C
C     CALCULATION OF THE TOTAL EMISSION SPECTRUM
C     ******************************************
C
      KMAX=INT((LMAX-LMIN)/STEP3)
      DO 120 K=0,KMAX,1
      DO 110 I=1,VIB,1
      SPUIT(K)=SPUIT(K)+P(I)*SPEK(I,K)
  110 CONTINUE
  120 CONTINUE
C
      WRITE(5,/) LMIN,LMAX,STEP3
      WRITE(5,/) SPUIT
C
      CLOSE(5,DISP=CRUNCH)
      END
```

MINIB2/S/GEBONDENVRIJ/PLOTTER
DATE & TIME PRINTED: MONDAY, JULY 20, 1987 @ 10:50:14.

```
C     MINIB2/S/GEBONDENVRIJ/PLOTTER
C
C     This program is part of a series of programs to simulate
C     bound-free rare gas halide emission spectra.
C     With this program a previewfile is created which gives a
C     graphical representation of an emission spectrum stored
C     on a specified FILE 1.
C
C*******************************************************************
C
FILE  1(KIND=DISK, FILETYPE=7, TITLE="MINIB2/D/GEBONDENVRIJ/POTUITVOER")
      BLOCK GLOBALS
 $ INCLUDE "PLOTTER/FORTRAN/DECLARATION ON APPL"
      END
 $ INCLUDE "PLOTTER/FORTRAN/POLYD ON APPL"
      REAL SPUIT,HOR,STEP3
      INTEGER LMIN,LMAX,I
      DIMENSION HOR(0:1019),SPUIT(0:1019)
C
      READ(1,/) LMIN,LMAX,STEP3
      READ(1,/) SPUIT
C
      KMAX=INT((LMAX-LMIN)/STEP3)
      DO 10 I=0,KMAX
      HOR(I)=LMIN+STEP3*I
   10 CONTINUE
C
      CALL POLYD(90,0,0,25,15,1,KMAX+1,HOR,SPUIT,.FALSE.,.FALSE.)

      STOP
      END
```

```
MINIB2/S/GEBONDENVRIJ/CONVERTER
DATE & TIME PRINTED: MONDAY, JULY 20, 1987 @ 10:50:29.
```

```
%       MINIB2/S/GEBONDENVRIJ/CONVERTER
%
%       This program is part of a series of programs to simulate
%       bound-free rare gas halide emission spectra.
%       With this program a data-file which has been transported
%       to the Burroughs 7900 system, can be converted to a file
%       D/DATAOUT, which is suitable for use in Fortran-programs.
%
%**********************************************************************
%
BEGIN
%
$INCLUDE "(TNNDAW20)LIBRARY/TOF/PREFIX ON USER4."
$INCLUDE "(TNNDAW20)MINIB2/S/EDGAR/ROTATIE/PROCS ON USER4."
INTEGER IDNR,IEERST,ILAATST,LMIN,LMAX;
REAL METING,STEP2;
ARRAY DATA[0:1019];
%
FILE LINEOUT(KIND=DISK,UNITS=CHARACTERS,MAXRECSIZE=132,PROTECTION=SAVE,
       NEWFILE=TRUE,TITLE="MINIB2/D/GEBONDENVRIJ/OUT ON POOL."),
     OUT(KIND=DISK,UNITS=CHARACTERS,MAXRECSIZE=132,PROTECTION=SAVE,
       NEWFILE=TRUE,TITLE="MINIB2/D/GEBONDENVRIJ/DATAOUT ON USER4."),
     DUMP(KIND=DISK,PACKNAME="POOL.",NEWFILE=TRUE,PROTECTION=SAVE,
       TITLE="DUMP.");
%
IDNR:=86721141;
METINGOPHALEN(IDNR,IEERST,ILAATST,DATA,LMIN,STEP2,DUMP);
LMAX:=(LMIN+(ILAATST-IEERST)*STEP2+.01);
WRITE(OUT,/,LMIN,LMAX,STEP2);
WRITE(OUT,/,DATA);
%
CLOSE(OUT,CRUNCH);
END.
```

MINIB2/S/GEBONDENVRIJ/CONVERTER/MAT
DATE & TIME PRINTED: MONDAY, JULY 20, 1987 @ 10:50:45.

```
%      MINIB2/S/GEBONDENVRIJ/CONVERTER/MAT
%
%      This program is part of a series of programs to simulate
%      bound-free rare gas halide emission spectra.
%      With this program a number of data-files which have been
%      transported to the Burroughs 7900 system, can be converted
%      into one file D/DATAOUT, which is suitable for use in Fortran
%      programs. The structure of the program is such that all the
%      the measurements at one wavelength are stored together.
%      This program should be used together with the programs
%      MINIB2/S/GEBONDENVRIJ/VOLGORDE and MINIB2/S/GEBONDENVRIJ/
%      OPTELLEN
%
BEGIN
%
$INCLUDE "(TNNDAW20)LIBRARY/TOF/PREFIX ON USER4."
$INCLUDE "(TNNDAW20)MINIB2/S/EDGAR/ROTATIE/PROCS ON USER4."
INTEGER I,J,K,IDNR,IEERST,ILAATST,LMIN,LMAX;
REAL METING,STEP2;
ARRAY DATA[0:450],IDN[1:50],MATRIX[1:50,0:450],DATA2[1:50];
%
FILE LINEOUT(KIND=DISK,UNITS=CHARACTERS,MAXRECSIZE=132,PROTECTION=SAVE,
       NEWFILE=TRUE,TITLE="MINIB2/D/GEBONDENVRIJ/OUT ON POOL."),
       OUT(KIND=DISK,UNITS=CHARACTERS,MAXRECSIZE=132,PROTECTION=SAVE,
       NEWFILE=TRUE,TITLE="MINIB2/D/GEBONDENVRIJ/DATAOUT ON USER4."),
       DUMP(KIND=DISK,PACKNAME="POOL.",NEWFILE=TRUE,PROTECTION=SAVE,
       TITLE="DUMP.");
  FOR I:=1 STEP 1 UNTIL 25 DO IDN[I]:=87651509+I;
%
  FOR I:=1 STEP 1 UNTIL 25 DO BEGIN
    IDNR:=IDN[I];
    METINGOPHALEN(IDNR,IEERST,ILAATST,DATA,LMIN,STEP2,DUMP);
    LMAX:=(LMIN+(ILAATST-IEERST)*STEP2+.01);
    FOR J:=1 STEP 1 UNTIL 450 DO MATRIX[I,J]:=DATA[J];
  END;
%
  WRITE(OUT,/,LMIN,LMAX,STEP2);
  FOR I:=1 STEP 1 UNTIL 450 DO
  BEGIN FOR J:=1 STEP 1 UNTIL 25 DO DATA2[J]:=MATRIX[J,I];
    WRITE(OUT,/,DATA2);
  END;
%
  CLOSE(OUT,CRUNCH);
END.
```

MINIB2/S/GEBONDENVRIJ/VOLGORDE
DATE & TIME PRINTED: MONDAY, JULY 20, 1987 @ 10:51:02.

```
C     MINIB2/S/GEBONDENVRIJ/VOLGORDE
C
C     This program is part of a series of programs to simulate
C     bound-free rare gas halide emission spectra.
C     This program takes measurements at one wavelength, which
C     have been stored by program MINIB2/S/GEBONDENVRIJ/CONVERTER/MAT,
C     and puts them in order of magnitude, so they are suitable for
C     program MINIB2/S/GEBONDENVRIJ/OPTELLEN.
C
C**********************************************************************
C
FILE  1(KIND=DISK, FILETYPE=7, TITLE="MINIB2/D/GEBONDENVRIJ/DATAOUT")
FILE  2(KIND=DISK, TITLE="MINIB2/D/GEBONDENVRIJ/DATAOUT/2",
     *  NEWFILE=TRUE, PROTECTION=SAVE)
      REAL STEP3
      INTEGER LMIN,LMAX,I,J
      DIMENSION DATA(1:50),DATA2(1:50)
C
      READ(1,/) LMIN,LMAX,STEP3
      WRITE(2,/) LMIN,LMAX,STEP3
C
      DO 100 L=1,450
C
      READ(1,/) DATA
      DATA2(1)=DATA(1)
      DO 50 I=2,25
      DO 40 J=1,I-1
      JAC=I-J
      IF(DATA(I).LE.DATA2(JAC)) GOTO 30
      DATA2(JAC+1)=DATA(I)
      GOTO 50
   30 DATA2(JAC+1)=DATA2(JAC)
      IF(JAC.EQ.1) DATA2(1)=DATA(I)
   40 CONTINUE
   50 CONTINUE
      WRITE(2,/) DATA2
C
  100 CONTINUE
C
      CLOSE(2,DISP=CRUNCH)
      STOP
      END
```

```
MINIB2/S/GEBONDENVRIJ/OPTELLEN
DATE & TIME PRINTED: MONDAY, JULY 20, 1987 @ 10:51:20.
```

```
C     MINIB2/S/GEBONDENVRIJ/OPTELLEN
C
C     This program is part of a series of programs to simulate
C     bound-free rare gas halide emission spectra.
C     In this program data stored by MINIB2/S/GEBONDENVRIJ/VOLGORDE
C     on a FILE 1, are added up. Because at each wavelength all
C     measurements have been stored in order of magnitude, noise
C     peaks can easily be removed.
C
C**********************************************************************

FILE 1(KIND=DISK,FILETYPE=7,TITLE="MINIB2/D/GEBONDENVRIJ/DATAOUT/2")
FILE 2(KIND=DISK, TITLE="MINIB2/D/GEBONDENVRIJ/TOTAAL",
     *  NEWFILE=TRUE, PROTECTION=SAVE)
      REAL SPUIT,HOR,STEP3
      INTEGER LMIN,LMAX,I
      DIMENSION DATA2(1:50),MATRIX(1:50,0:500),TOTAAL(0:1019)
C
      READ(1,/) LMIN,LMAX,STEP3
      WRITE(2,/) LMIN,LMAX,STEP3
C
      DO 10 I=0,449
      READ(1,/) DATA2
      DO 5 J=1,22
      TOTAAL(I)=TOTAAL(I)+DATA2(J)
  5   CONTINUE
  10  CONTINUE
      WRITE(2,/) TOTAAL
C
      CLOSE(2,DISP=CRUNCH)
C
      STOP
      END
```

MINIB2/S/GEBONDENVRIJ/POTPLOTTER
DATE & TIME PRINTED: MONDAY, JULY 20, 1987 @ 10:52:12.

```
C     MINIB2/S/GEBONDENVRIJ/POTPLOTTER
C
C     This program is part of a series of programs to simulate
C     bound-free rare gas halide emission spectra.
C     With this program a graphical representation can be given of
C     a set of potential curves and Mulliken's difference potential,
C     according to specifications on a FILE 1.
C
C*******************************************************************
C
FILE  1(KIND=DISK,FILETYPE=7,TITLE="MINIB2/D/GEBONDENVRIJ/POTINVOERCA")
      BLOCK GLOBALS
 $ INCLUDE "PLOTTER/FORTRAN/DECLARATION ON APPL"
      END
 $ INCLUDE "PLOTTER/FORTRAN/POLYDZ ON APPL"
      REAL H,RMIN,RMAX,UPPARR,LOWARR,R,UUP,ULO,E,UMUL
      INTEGER I,N
      DIMENSION UPPARR(9),LOWARR(9),R(0:4001),UUP(0:4001),ULO(0:4001),
     *   UMUL(0:4001)
C
      READ(1,/) DUMMY
      READ(1,/) DUMMY
      READ(1,/) RMIN,RMAX
      READ(1,/) N
      READ(1,/) DUMMY,DUMMY
      READ(1,/) DUMMY,DUMMY
      READ(1,/) DUMMY,DUMMY
      READ(1,/) DUMMY,DUMMY
      READ(1,/) DUMMY
      READ(1,/) UPPARR(1),
     *   UPPARR(2),UPPARR(3),UPPARR(4),UPPARR(5),UPPARR(6),
     *   UPPARR(7),UPPARR(8),UPPARR(9)
      READ(1,/) LOWARR(1),
     *   LOWARR(2),LOWARR(3),LOWARR(4),LOWARR(5),LOWARR(6),
     *   LOWARR(7),LOWARR(8),LOWARR(9)
      READ(1,/) DUMMY
      READ(1,/) DUMMY,DUMMY,DUMMY,DUMMY
      READ(1,/) DUMMY
      READ(1,/) DUMMY
      READ(1,/) E
C
      H=(RMAX-RMIN)/N
      DO 30 I=0,N+1,1
      R(I)=RMIN+I*H
      UUP(I)=UPPARR(1)+UPPARR(2)
     *   +UPPARR(3)*EXP(-UPPARR(4)*R(I))
     *   +UPPARR(5)*EXP(-UPPARR(6)*R(I))
     *   -UPPARR(7)/R(I)
     *   -UPPARR(8)/(R(I)**4)
     *   -UPPARR(9)/(R(I)**6)
      ULO(I)=LOWARR(1)+LOWARR(2)
     *   +LOWARR(3)*EXP(-LOWARR(4)*R(I))
     *   +LOWARR(5)*EXP(-LOWARR(6)*R(I))
     *   -LOWARR(7)/R(I)
     *   -LOWARR(8)/(R(I)**4)
     *   -LOWARR(9)/(R(I)**6)
      UMUL(I)=ULO(I)+E-UUP(I)
  30  CONTINUE
```

```
C
      CALL POLYDZ(90,0,0,20,25,1,N+2,R,UUP,10,1.5,6.5,12,0,60000,
     *   .FALSE.,.FALSE.)
      CALL POLYDZ(90,0,0,20,25,1,N+2,R,ULO,10,1.5,6.5,12,0,60000,
     *   .FALSE.,.FALSE.)
      CALL POLYDZ(90,0,0,20,25,1,N+2,R,UMUL,10,1.5,6.5,12,0,60000,
     *   .FALSE.,.FALSE.)
C
      STOP
      END
```

MINIB2/S/GEBONDENVRIJ/OMKEERPLOT
DATE & TIME PRINTED: MONDAY, JULY 20, 1987 @ 10:51:38.

```
C     MINIB2/S/GEBONDENVRIJ/OMKEERPLOT
C
C     This program is part of a series of program to simulate
C     bound-free rare gas halide emission spectra.
C     In this program, for a number of sets of potential curves
C     specified, a figure is plotted giving the wavelengths of
C     emissions from the classical turning points as a function
C     of vibrational energy.
C
C**********************************************************************
C
C     DECLARATIONS
C     ************
C
FILE  2(KIND=REMOTE)
FILE  3(KIND=PRINTER)
FILE  4(KIND=DISK,FILETYPE=7,TITLE="MINIB2/D/GEBONDENVRIJ/POTINVOERCA")
      BLOCK GLOBALS
 $ INCLUDE "PLOTTER/FORTRAN/DECLARATION ON APPL"
      END
 $ INCLUDE "PLOTTER/FORTRAN/POLYDZ ON APPL"
      REAL STEP3,STEP,UMINUP,EU,SPEK,DUMMY,RMIN,RMAX,UPPARR,
     *   LOWARR,H,UDIF,R,UUP,ULO,ULO2,UMUL,UMUL2,REUP,GAMMA,GAMMA2,
     *   D1,D2,D3,D4,D5,C1,C2,C3,C4,C5,TM,TM2,RBI,LABBI1,LABBI2,
     *   LABMIN,RBU,LABMAX,G,LAB,NU,R1,LABIAC,LABUAC,RKOUD,
     *   R2,DELTTE,CORR,LABBI,LABBU,LABOUD,ROUD,RNU,NUOUD,LOUD
      INTEGER LMIN,LMAX,KMAX,VIB,N,IREUP,IBI,IBU,I1,I2,JBIN,
     *   JBUI,IOUD,SPEK2,INU,IKOUD
      DIMENSION EU(0:179),SPEK(0:29,0:1019),UPPARR(1:9),
     *   LOWARR(1:9),UUP(0:1001),ULO(0:1001),UMUL(0:1001),
     *   TM(0:1001),TM2(0:1001),G(0:1001),R(0:1001),
     *   IR1(0:1001),IR2(0:1001),IBI(0:179),IBU(0:179),
     *   SPEK2(0:29,0:1019),UMUL2(0:1001),ULO2(0:1001),
     *   RULO(0:9),UULO(0:9)
C
C     INPUT VARIABLES
C     ***************
C
      READ(4,/) DUMMY
      READ(4,/) DUMMY
      READ(4,/) RMIN,RMAX
      READ(4,/) N
      READ(4,/) DUMMY,DUMMY
      READ(4,/) DUMMY,DUMMY
      READ(4,/) DUMMY,DUMMY
      READ(4,/) DUMMY,DUMMY
      READ(4,/) DUMMY
      READ(4,/) UPPARR(1),
     *   UPPARR(2),UPPARR(3),UPPARR(4),UPPARR(5),UPPARR(6),
     *   UPPARR(7),UPPARR(8),UPPARR(9)
      READ(4,/) LOWARR(1),
     *   LOWARR(2),LOWARR(3),LOWARR(4),LOWARR(5),LOWARR(6),
     *   LOWARR(7),LOWARR(8),LOWARR(9)
C
C     CALCULATION THE VIBRATIONAL ENERGY DEPENDENCE OF THE WAVELENGTHS
C     OF THE TURNING POINT EMISSION FOR A SPECIFIED NUMBER OF
C     POTENTIAL CURVES
C     ****************************************************************
```

```
C
      H=(RMAX-RMIN)/(N+1)
      DO 32 K=1,8

      UMINUP=10**10
      UDIF=10**10
      DO 31 I=0,N+1,1
      R(I)=RMIN+I*H
      UUP(I)=UPPARR(1)+UPPARR(2)
     *   +UPPARR(3)*EXP(-UPPARR(4)*R(I))
     *   +UPPARR(5)*EXP(-UPPARR(6)*R(I))
     *   -UPPARR(7)/R(I)
     *   -UPPARR(8)/(R(I)**4)
     *   -UPPARR(9)/(R(I)**6)
      ULO(I)=LOWARR(1)+LOWARR(2)
     *   +LOWARR(3)*EXP(-LOWARR(4)*R(I))
     *   +LOWARR(5)*EXP(-LOWARR(6)*R(I))
     *   -LOWARR(7)/R(I)
     *   -LOWARR(8)/(R(I)**4)
     *   -LOWARR(9)/(R(I)**6)
C
      IF(K.EQ.1) ULO2(I)=ULO(I)
      IF(K.EQ.2) ULO2(I)=4.4433E7*EXP(-2.92*R(I))+700
      IF(K.EQ.3) ULO2(I)=5.6438E7*EXP(-3.02*R(I))
      IF(K.EQ.4) ULO2(I)=3.7985E7*EXP(-2.85*R(I))
      IF(K.EQ.5) ULO2(I)=2.6877E7*EXP(-2.7*R(I))
      IF(K.EQ.6) ULO2(I)=6.2258E7*EXP(-3.0*R(I))
      IF(K.EQ.7) ULO2(I)=8.2376E7*EXP(-3.1*R(I))
      IF(K.EQ.8) ULO2(I)=1.0900E8*EXP(-3.2*R(I))
C
      UMUL2(I)=1E7/(UUP(I)-ULO2(I))
      UUP(I)=UUP(I)-UPPARR(1)
   31 CONTINUE
C
      CALL POLYDZ(90,0,0,25,20,1,N+1,UMUL2,UUP,5,250,400,5,0,20000,
     *   .FALSE.,.FALSE.)
C
   32 CONTINUE
C
      CLOSE(6,DISP=CRUNCH)
      END
```

MINIB2/S/GEBONDENVRIJ/FITFUNCPLOT
DATE & TIME PRINTED: THURSDAY, JULY 30, 1987 @ 14:32:21.

```
C      MINIB2/S/GEBONDENVRIJ/FITFUNCPLOT
C
C      This program is part of a series of programs to simulate
C      bound-free rare gas halide emission spectra.
C      With this program a graphical representation can be given of
C      the fit-functions used in MINIB2/S/GEBONDENVRIJ/VIBFIT and
C      MINIB2/S/GEBONDENVRIJ/VIBFIT/TRIPEL.
C
C*****************************************************************
C
FILE  1(KIND=DISK,TITLE="MINIB2/D/GEBONDENVRIJ/XEBRCA/5")
      BLOCK GLOBALS
 $ INCLUDE "PLOTTER/FORTRAN/DECLARATION ON APPL"
      END
 $ INCLUDE "PLOTTER/FORTRAN/POLYDZ ON APPL"
      INTEGER LMIN,LMAX,VIB,STEP
      REAL STEP3,UMINUP,EU,EU2,SPEK,SPEK2,DATA,HOR
      DIMENSION SPEK(0:9,0:1019),EU(0:179),DATA(0:1019),HOR(0:1019)
C
      READ(1,/) LMIN,LMAX,STEP3
      READ(1,/) VIB,STEP
      READ(1,/) UMINUP
      READ(1,/) EU
C
      DO 20 I=0,VIB-1
      DO 10 J=1,34
      READ(1) (SPEK(I,30*(J-1)+K),K=0,29)
   10 CONTINUE
   20 CONTINUE
C
      DO 22 I=0,VIB-1
      DO 21 J=0,1019
   21 IF(SPEK(I,J).GT.SPEKMA) SPEKMA=SPEK(I,J)
   22 CONTINUE
C
      KMAX=INT((LMAX-LMIN)/STEP3)
      DO 30 I=0,VIB-1
      DO 25 J=0,KMAX
      HOR(J)=LMIN+J*STEP3
   25 DATA(J)=SPEK(I,J)/SPEKMA+.5*I
      CALL POLYDZ(90,0,0,25,20,1,KMAX+1,HOR,DATA,5,LMIN,LMAX,
     *   VIB,0,.5*VIB+.5,.FALSE.,.FALSE.)
   30 CONTINUE
C
      END
```