$$\mathbf{D}^{\eta}_{h,i} = \begin{pmatrix} \mathbf{D}^{d}_{r\ d,b} & \mathbf{D}^{d}_{\alpha\ d,a} \\ \mathbf{O}_{a,b} & \mathbf{E}_{a,a} \end{pmatrix} r = r_1, r_2, \ldots, r_b \qquad (31)$$

$$\mathbf{D}^{\xi}_{i,i} = \begin{pmatrix} \mathbf{E}_{b,b} & \mathbf{O}_{b,a} \\ \mathbf{D}^{r'}_{r\ a,b} & \mathbf{D}^{r'}_{\alpha\ a,a} \end{pmatrix} \begin{matrix} r = r_1, r_2, \ldots, r_b \\ r' = r_{b+1}, r_{b+2}, \ldots r_i \end{matrix} \qquad (32)$$

where **E** and **O** are the indentity and null matrices.

We wish to thank Prof. Otto Bastiansen and Mr Fred N. Fritsch for helpful discussion, and for significant contribution to the **J** matrix calculation.

References

ARLEY, N. & BUCH, K. R. (1950). *Introduction to the Theory of Probability and Statistics.* Chapter 12. New York: Wiley.

BASTIANSEN, O., HEDBERG, L. & HEDBERG, K. (1956). *J. Chem. Phys.* **27**, 1311.

BONHAM, R. A. & BARTELL, L. S. (1959). *J. Chem. Phys.* **31**, 702.

CRUICKSHANK, D. W. J. & VIERVOLL, H. (1949). *Acta Chem. Scand.* **3**, 560.

HAMILTON, W. A. & SCHOMAKER, V. (1954). Unpublished. But see Hamilton, W. A., Ph D. Thesis, California Institute of Technology, 1955.

HUGHES, E. W. (1941). *J. Amer. Chem. Soc.* **63**, 1737.

Acta Cryst. (1964). **17**, 533

Least-Squares Refinement of Molecular Structures from Gaseous Electron-Diffraction Sector-Microphotometer Intensity Data. II. Adaptation to Automatic Computation*

BY MACHIO IWASAKI,† F. N. FRITSCH AND KENNETH HEDBERG‡

Department of Chemistry, Oregon State University, Corvallis, Oregon, U.S.A.

(*Received* 1 *May* 1963)

A description is given of a computer program representing an adaptation of a least-squares method of molecular structure refinement based on gaseous electron-diffraction sector-microphotometer data.

Introduction

The principles of a least-squares refinement method, based on gaseous electron-diffraction sector-microphotometer data, were described in the preceding article (Hedberg & Iwasaki, 1964; hereafter called H & I). In a typical application of the method some 200 observations and 10–15 parameters may be handled, and the calculations involved are extensive enough to merit use of an automatic computer. Because the power of the method should attract increasing attention from workers in gaseous electron diffraction, and because the computational problems posed are quite different from those encountered in X-ray diffraction, a description of a computer program seems appropriate.

In the account given here we seek to present only the main features of the program and in a way to make clear the sequence of operations. The actual program, which will be of little interest to others since it was written for a non-standard ALWAC III-E, differs in unimportant details from the flow diagrams. The complexity of the program is in part dictated by the nature of the computational problem and in part by the properties of the rather slow computer superposed on our objectives of making the refinement as automatic as possible and as versatile as the basic intensity equation H & I (14) permits. Thus, it has been necessary to build in special features such as optional approximations which allow savings of time, or checking calculations with optional recycling to protect a large investment of time. Use of one of the widely available high-speed computers, where a refinement cycle for the typical problem mentioned above would require seconds or minutes instead of 1–1½ hours as it does with ALWAC III-E, would make these features unnecessary and lead to simplifications.

List of symbols

For further explanation of many of the following quantities see H & I.

A Matrix of derivatives.
$\mathbf{A} = \mathbf{A}_{Nm} = \{\partial k I_t^{\mathrm{calc}}(s)/\partial x_i\}_{Nm}$

B $\mathbf{B} = \mathbf{A}'\mathbf{P}\mathbf{A}$.

* This work was supported by the Directorate of Chemical Sciences, Air Force Office of Scientific Research, under contract AF 49 (638)–783.

† Present address: Government Industrial Research Institute, Nagoya, Hirate-machi, Kita-ku, Nagoya, Japan.

‡ Alfred P. Sloan Foundation Fellow.

$\mathbf{D}_\chi^\xi$, $\mathbf{D}_\chi^\eta$, $\mathbf{D}_r^\alpha$ Matrices used in calculation of **J**. See Appendix of H & I.

$\mathbf{F}^{obs}$ Matrix of observed values of function. $\mathbf{F}^{obs}=\{I^{obs}(s)\}_{N1}$.

$\mathbf{F}^\circ$ Matrix of function corresponding to trial values of parameters. $\mathbf{F}^\circ=\{kI_0^{calc}(s)\}_{N1}$.

F Matrix of function corresponding to adjusted values of parameters. $\mathbf{F}=\mathbf{F}_N=\{kI^{calc}(s)\}_{N1}$.

J Matrix expressing dependence of non-parameter interatomic distances on distances chosen as parameters. $\mathbf{J}=\{\partial d_d/\partial r_i\}_{di}$.

N $\mathbf{N}=\mathbf{F}^{obs}-\mathbf{F}=\{I^{obs}(s)-kI_0^{calc}(s)\}_{N1}$.

P Diagonal weight matrix. $\mathbf{P}=\{P_{ij}\}_{NN}$; $P_{ij}\begin{cases}=0 & i\neq j\\ =P_{ii} & i=j\end{cases}$.

X Matrix of corrections to trial values of parameters.

Y $\mathbf{Y}=\mathbf{A}'\mathbf{P}\mathbf{N}$.

$\sigma(\mathbf{X}_q)$ Standard error. $\sigma(\mathbf{X}_q)=[\mathbf{B}_{qq}^{-1}(\mathbf{V}'\mathbf{P}\mathbf{V})/(N-m)]^{\frac{1}{2}}$.

$\Delta\eta_{ij}$ Phase coefficient for atomic pair i,j.

A_n Scattered amplitude from nth type of symmetry equivalent atomic pairs. $A_n=A_0$ or $A_n=A_{ij}$.

A_0 Constant scattered amplitude.

(A_0) Vector of values of constant amplitudes.

A_{ij} Variable amplitude. $A_{ij}=n'Z_iZ_jF_iF_j\cos\Delta\eta_{ij}$.

(A_{ij}) Matrix of values of variable amplitudes.

b Arbitrary constant used in calculation of **P**.

d Number of interatomic distances not parameters. $d=n-i$.

d_d Interatomic distances not parameters.

e Number of different kinds of atoms.

F_k Atomic coefficient of atom k normalized to an arbitrary reference atom M. $F_k=(Z_k-f_k)(Z_M-f_M)^{-1}Z_MZ_k^{-1}$.

$I_t^{calc}(s)$ Calculated scattered intensity. $I_t^{calc}(s)=\sum_n A_nr_n^{-1}\exp(-\frac{1}{2}l_n^2s^2)\sin sr_n$.

$I_n^{calc}(s)$ Scattered intensity arising from a single type of symmetry-equivalent atomic pairs. $I_n^{calc}(s)=A_nr_n^{-1}\exp(-\frac{1}{2}l_n^2s^2)\sin sr_n$.

$I^{obs}(s)$ Observed scattered intensity.

i Number of different interatomic distances taken as parameters.

(i,j) Vector of subscripts identifying a variable amplitude.

j Number of mean amplitudes of vibration taken as parameters.

(j) Vectors of subscripts designating mean amplitudes taken as parameters.

k Amplitude scale parameter. $k=[\sum_s I^{obs}(s).I^{calc}(s)]/\sum_s(I^{calc}(s)^2$.

l_n Root-mean-square amplitude (mean amplitude) of vibration of nth type of symmetry-equivalent atomic pairs.

l_i Mean amplitude corresponding to distance r_i.

(l_0) Vector of values of root-mean-square amplitudes (mean amplitudes).

m Total number of parameters. $m=i+j+1$.

N Number of observations.

n Number of different types of symmetry-equivalent atomic pairs or interatomic distances.

n' Number of symmetry-equivalent distances of a given type.

(n') Vector of values of n'.

r_n Interatomic distance of nth type of symmetry-equivalent atomic pairs.

r_i Interatomic distances taken as parameters.

(r_0) Vector of values of interatomic distances.

S $S=[(\mathbf{V}'\mathbf{P}\mathbf{V})/(N-m)]^{\frac{1}{2}}$.

s Electron diffraction variable. $s=4\pi\lambda^{-1}\sin\theta$.

Features of the program

The main features of the program are the following.

1. Any number of parameters up to 30 may be refined simultaneously.

2. The amplitudes A_n may be constants or vary with s.

3. Molecules with up to five different atoms may be refined using variable amplitudes A_n (any number with constant amplitudes).

4. The refinement may be based on any continuous range of s and any interval Δs.

5. The starting value of the scale parameter k may be calculated automatically or a trial value read.

6. Any number up to $(n-1)$ of the mean amplitudes l_n may be held constant.

7. The diagonal weight matrix may be read in, or it may be calculated according to the formula $P=\mathcal{N}s\exp(-bs^2)$, where b is an arbitrary constant and $\mathcal{N}$ a normalization constant.

8. The **J** matrix may be read in or calculated from bond-distance and bond-angle parameters.

9. Correct calculation of **B** is assured by an automatic sum-check which initiates recalculation in case of machine error. A 'manual' sum-check in place of the automatic sum-check is optional.

10. The $\mathbf{B}^{-1}$ matrix from a previous refinement cycle may be used optionally.

11. Inputs are $\mathbf{F}^{obs}$, **P** or b, s_{min}, s_{max}, Δs, k (optional), n, i, j, (r_0), (l_0), and (j) if $j\neq n$, for all refinements. Additional inputs are **J**, or a row vectors of $\mathbf{D}_\chi^\xi$, and d row vectors of $\mathbf{D}_\chi^\eta$ if **J** is to be calculated; and (A_0) or e, (n'), (i,j) and (A_{ij}) depending upon whether A_n in $I_t^{calc}(s)$ are constants or vary with s.

12. Outputs are $\mathbf{F}^{obs}$ (optional), **P** (optional), **J** (optional), k, (r_0), (l_0), $\mathbf{D}_r^\alpha$ (when **J** is calculated), **A**

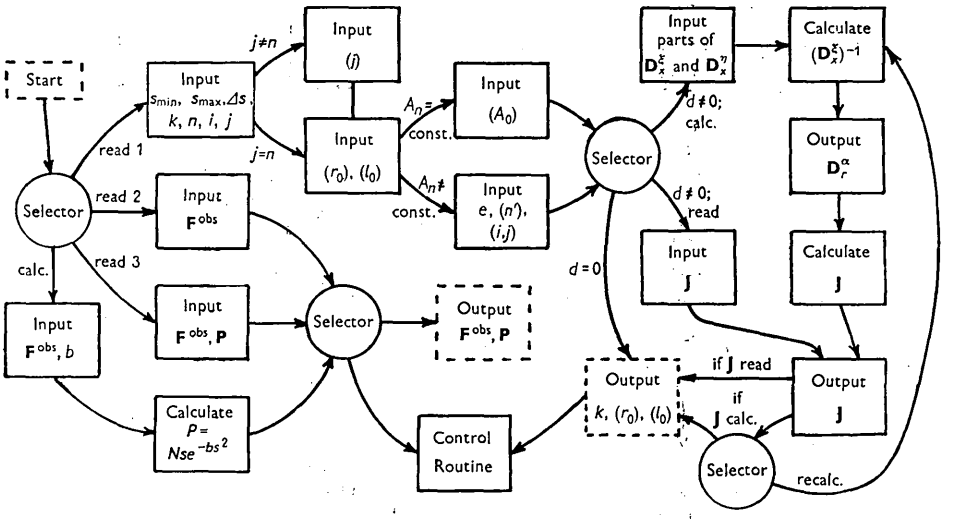

Fig. 1. Input routine flow diagram.
Selectors indicate operator choices, branched lines from a box indicate an internal choice.
Dashed boxes indicate operations actually carried out in other routines.

(optional), sum-checks (if 'manual' was elected), **N'PN**, **A'PN**, **B**, **BX**, **V'PV**, S, $\mathbf{B}^{-1}$, $\mathbf{X} \pm \sigma(\mathbf{X})$, (r_i), (d_d), (l), and **F** (optional).

Program routines

The program consists of four routines: input, control, derivative, and least-squares. The input routine is concerned principally with data storage, and the calculation of **P** and **J**. The control routine coordinates the operations of the other routines, calculates distances and mean amplitudes for the corrected structure, and arranges for outputs. The derivatives routine calculates k, $\mathbf{F}^{calc}$, **A**, and quantities to be used in the sum-check. The least-squares routine calculates the sum-check, forms the normal equations, and solves for **X** and $\sigma(\mathbf{X})$. A description of the more important functions of each routine follows.

Input routine

It is with this routine that the refinement problem is defined. Choices to be made are: (1) Which (if any) of the l_n are not to be parameters? (2) Are the A_n to be constants or variables? (3) Is a **P** matrix different from unity needed, and if so is it to calculated by the standard formula or read? (4) Is a **J** matrix needed, and if so is it to be calculated or read? Decisions regarding these are made by combinations of manually operated selector switches and code words accompanying the data, and occasionally by the nature of the data themselves. For example, as may be seen from Fig. 1, the first choice above is decided by a selector set to read s_{min}, s_{max}, *etc.* and by j and n; an input of the parameters of the starting model is next requested, followed by a request for input of (j) if $j \neq n$.

In addition to defining the type of refinement this routine arranges for data storage and calculates **P** and **J**, if necessary. The method of the **J** matrix calculation is found in the appendix of H & I.

Control routine

The functions of this routine are seen from Fig. 2. Observed data are handled in blocks of 31. Derivatives are first calculated, after which the results are conveyed to the least-squares routine where components of **B** are obtained from them. This process, repeated

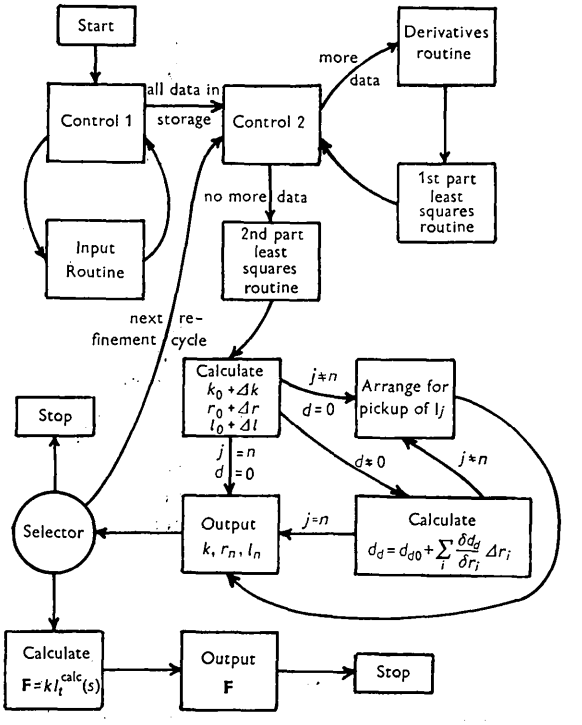

Fig. 2. Control routine flow diagram.

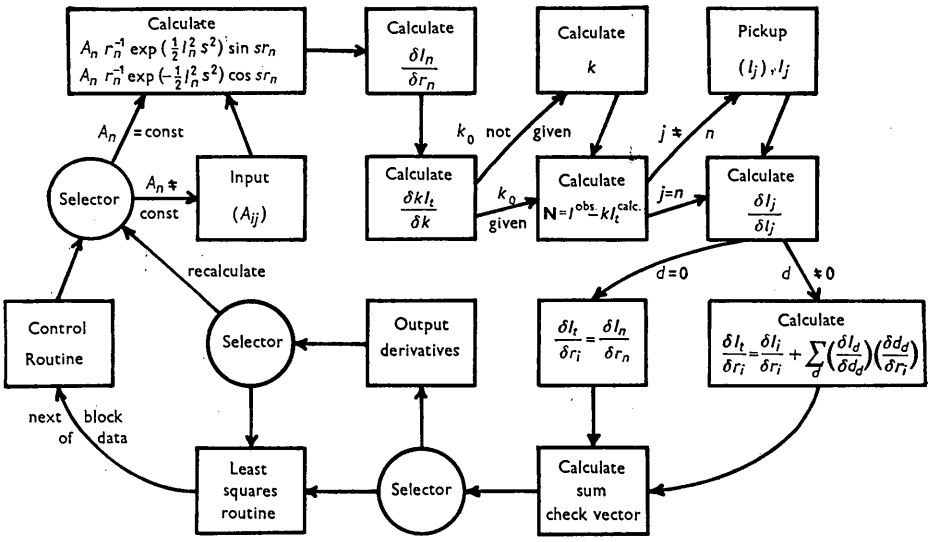

Fig. 3. Derivatives routine flow diagram. Selectors indicate operator choices, branched lines from a box indicate an internal choice.

for the remaining data, is indicated by the first three boxes in the figure. When the last block is complete and the normal equations solved, the new distances and mean amplitudes are calculated in preparation for a new cycle.

Derivatives routine

The primary function of this routine is the calculation of **A**. The flow of operations is seen from Fig. 3. The routine begins with calculation of the $\exp \times \sin sr_n$ and $\exp \times \cos sr_n$ functions using constant or variable A_n (the principles of the variable A_n calculation are briefly described in Appendix I): the exponential part is done first, so that when it becomes insignificant the remainder of the calculation can be skipped. The derivatives $\partial I_n/\partial r_n$ are obtained from these (H & I (18)) as well as $\partial k I_t/\partial k = I_t$. If a trial value of k was not supplied to the input routine it is calculated before forming **N**. The derivatives $\partial I_j/\partial l_j$ are next calculated; these are simply $\partial I_n/\partial l_n$ if no l_n are held constant, otherwise the l_n to be refined (l_j) and the corresponding I_j are first picked up using (j) (Fig. 1). The calculation of $\partial I_t/\partial r_i$ follows a somewhat similar scheme: If $d=0$ these are given by the $\partial I_n/\partial r_n$ obtained in the first step of the routine; if $d \neq 0$ the **J** matrix must be employed (Fig. 1). After calculation of the sum-check vector a decision for output of derivatives is required; if an output is elected a recalculation option is available.

Least-squares routine

The flow diagram is given in Fig. 4. Two alternative paths are available depending upon whether **B** from the previous refinement cycle is to be used or not. In early refinement cycles path 1 is followed. Elements of **B**, **Y**, and **N'PN** are calculated and the sum-check (Appendix II) made for each block of data, the program alternating between this and the derivatives routine through the main control routine. Two sum-check options are available. With the manual subroutine a decision to recalculate (in case of an error) or to proceed with the next block of data is required; with the automatic subroutine the decision is made internally. After the final block of data has been handled **Y** and **N'PN** are output and the normal equations solved. Before output of $\mathbf{B}^{-1}$, **BX** and **Y** must be compared to check the matrix inversion and a decision made to refine $\mathbf{B}^{-1}$ or to proceed. The refinement cycle is completed with calculations and outputs of **V'PV**, $\mathbf{X} \pm \sigma(\mathbf{X})$, and the new parameter values. A new cycle may then be started, or an intensity curve corresponding to the refined structure taken out. Path 2, followed in late cycle of refinement, shortens the time required. In this path the derivatives are calculated as before, followed by **Y** and **N'PN**; outputs of **B** and $\mathbf{B}^{-1}$ are omitted.

APPENDIX I

When variable amplitudes A_{ij} are to be used it is necessary to prepare a table of (A_{ij}), blocks of which are read in as needed. The table is prepared for each combination of different atom pairs (for example all compounds of C, H and O use the same table); if wished, this may be done by a separate routine from sub-master tapes of F_k, themselves prepared from master tapes of f_k. (A_{ij}) may be thought of as a matrix of $e(e+1)$ rows and N columns, blocks of

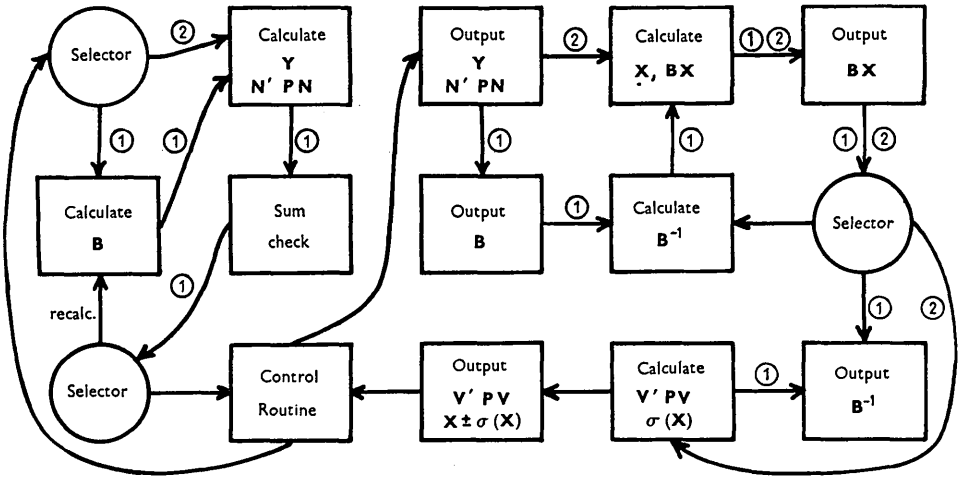

Fig. 4. Least-squares routine flow diagram. Paths 1 and 2 are alternative options. Selectors indicate operator choices.

which are read in as needed. Each block consists of 31 data for each atom pair type. The derivatives routine selects, using (i, j), the appropriate data associated with a given r_n when needed. This vector of subscripts consists of n elements and is determined by the order in which the F_k tapes were read in to form the table of (A_{ij}). Thus, if F_k for C, H, and O were used in that order, all C $\cdots$ C distances are designated by 1, 1; all C $\cdots$ H by 1, 2 all H $\cdots$ H by 2, 3; etc. Appropriate weighting of the different terms is obtained from (n').

APPENDIX II

We designate by $\overline{\Sigma \mathbf{A}}$ a vector with elements consisting of the sum of elements in a row of **A**. If this column is used to augment **A**, premultiplication of the augmented matrix **U** gives an extra column $\mathbf{U}\overline{\Sigma \mathbf{A}}$ of which the kth element is

$$\sum_i u_{ki}\left(\sum_j a_{ij}\right) = \sum_j \sum_i u_{ki} a_{ij} .$$

Now, the kth element of a vector $\overline{\Sigma \mathbf{UA}}$ with elements consisting of the sum of elements in each row of **UA**, is

$$\sum_j \sum_i u_{ki} a_{ij}$$

and, hence

$$\mathbf{U}\overline{\Sigma \mathbf{A}} = \overline{\Sigma \mathbf{UA}} .$$

In the sum check calculation the matrix **U** above is **A'P**, and the comparison made is $\overline{\Sigma \mathbf{A'PA}}$ with $\mathbf{A'P}\overline{\Sigma \mathbf{A}}$.

Reference

HEDBERG, K. & IWASAKI, M. (1964). *Acta Cryst.* **17**, 529.