Acta Cryst. (1973). A**29**, 445

On the Fast Computing of Three-Dimensional Fourier Transforms of Crystallographic Data *via* External Storage

BY S. LANGE,* U. STOLLE AND G. HUTTNER

Anorganisch-chemisches Laboratorium der Technischen Universität München, Germany (BRD)

(*Received* 27 *December* 1972; *accepted* 26 *February* 1973)

Fast Fourier Transforms (FFT) are widely known as useful tools for the evaluation of spectral data. This article discusses the applicability of FFT methods to crystallographic problems. Formulae are derived which make it possible to use fast Fourier transforms for the general Fourier summation of crystallographic data in all space groups and for the computation of slant planes at arbitrary positions in the unit cell. As even moderate resolutions would produce arrays of data too large to fit within internal computer memory, they must be kept to an external storage device. The organization of data and the resulting time requirements are thoroughly discussed. An ALGOL 60 program has been developed with as many redundancies eliminated as possible. An example shows that for a problem of moderate size this algorithm is faster by an order of magnitude than those which have traditionally been used.

Introduction

Several years ago, Cooley & Tukey (1965) discovered the principle of the fast Fourier transform (FFT), an effective algorithm for the computation of expressions of the form:

$$W_k = \sum_{j=0}^{N-1} w_j \,.\, \exp{(i2\pi jk/N)}\,, \qquad (1)$$

with

$$j=0,\quad 1,\ldots,N-1$$

and

$$k=0,\quad 1,\ldots,N-1\,.$$

For the use of their program, the dimension N of the complex array to be transformed must be a power of 2; this requirement can always be achieved by filling up with zeros. The transformation is then done in steps building upon one another. Thus, the computing time drops from N^2 for the straightforward summation to $2N\log_2 N$. The gain increases rapidly with N; for $N=32$ the ratio is already $\frac{3}{1}$!

This powerful method stimulated the development of the frequency analysis of acoustic vibrations in an extraordinary manner (*IEEE Transactions*, 1967, 1969; Cooley, Lewis & Welch, 1969*a*). The algorithm itself was developed in several directions. Besides the radix-2 algorithms (Gentleman & Sande, 1966; Singleton, 1968*a*), mixed-radix algorithms recently appeared allowing for different values of N (Singleton, 1968*b*, 1969). In general, however, the former are still more favourable, as they can treat phase factors more effectively. Assembler coding provides for additional time saving in address manipulation.

Multidimensional transforms can be performed by FFT algorithms when the corresponding expressions can be split up into single sums (Singleton, 1969; Brenner, 1969). For instance, multiple sums of the type:

$$D(xyz) = \sum_{h=0}^{h_e}\sum_{k=0}^{k_e}\sum_{l=0}^{l_e} d_{hkl} \times \exp{[i2\pi(hx/N+ky/O+lz/P)]} \qquad (2)$$

can be split up into three single sums:

$$D(xyz) = \sum_{h=0}^{h_e}\exp{(i2\pi hx/N)}\,.\,\sum_{k=0}^{k_e}\exp{(i2\pi ky/O)} \times \sum_{l=0}^{l_e}\exp{(i2\pi lz/P)}\,.\,d_{hkl}\,. \qquad (2a)$$

Each of the single sums can be transformed by the FFT, the other indices remaining constant. Bondot (1971) describes a combination of this method and Cooley & Tukey's algorithm itself. He does not transform in each dimension separately; the single steps of the procedure are prccessed in all dimensions at the same time. Equal sets of phase factors are computed only once.

Examples for the application of the FFT to crystallographic computations are rare (see, *e.g.*, Hoppe, Gassman & Zechmeister, 1970), although the phase problem forces extensive use of Fourier transforms.

General procedure for the FFT of three-dimensional crystallographic data

In crystal-structure analysis, the general formula to be computed is the following:

$$\varrho(xyz) = \frac{1}{V}\,.\sum_{h=h_a}^{h_e}\sum_{k=k_a}^{k_e}\sum_{l=l_a}^{l_e} F_{hkl} \times \exp{[-i2\pi(hx/N+ky/O+lz/P)]}\,. \qquad (3)$$

In this equation, F_{hkl} is the Fourier coefficient indexed by integer numbers h, k and l; (h_a, h_e), (k_a, k_e), (l_a, l_e)

* Present address: Department of Chemistry, The University of Wisconsin, Madison, WI 53706, U.S.A.

are the limits of these indices; $\varrho(xyz)$ is the value of the transform at discrete coordinates x, y, and z. N, O, P are the number of subdivisions of each row in the three directions of the transformed space, and $1/V$ is a scaling factor.

The Beevers–Lipson technique (see *e.g.*, Rollett, 1965; Stout & Jensen, 1968) is widely used for the effective evaluation of these summations. Only one independent part of the set of Fourier coefficients has to be summed in each dimension; the other parts are implicitly provided for by multiplicity factors that can be derived from Friedel's law and symmetry relations.

For the application of equation (2), the indices of the Fourier coefficients must be positive. In general, however, crystallographic data are indexed both positively and negatively. By means of the linear transformations

$$h'=h-h_a\,, \tag{4a}$$

$$k'=k-k_a\,, \tag{4b}$$

and

$$l'=l-l_a \tag{4c}$$

the general expression for the Fourier summation (3) can be converted to the form:

$$\begin{aligned}\varrho(xyz)=\exp\,(-i2\pi h_ax/N)\,.\sum_{h'=0}^{h_e-h_a}\exp\,(-i2\pi h'x/N)\\ \times i\exp\,(-i2\pi k_ay/\mathrm{O}\,.\sum_{k'=0}^{k_e-k_a}\exp\,(-i2\pi k'y/0)\\ \times\exp\,(-i2\pi l_az/P)\,.\sum_{l'=0}^{l_e-l_a}\exp\,(-i2\pi l'z/P)\\ \times\frac{1}{V}\,.\,F_{h_a+h',\;k_a+k',\;l_a+l'}\,.\end{aligned} \tag{5}$$

This equation is more comprehensive if we introduce:

$$F^0_{h'k'l'}=\frac{1}{V}F_{h_a+h',\;k_a+k',\;l_a+l'}=\frac{1}{V}F_{hkl}\,, \tag{6}$$

$$F'_{h'k'z}=\exp\,(-i2\pi l_az/P)\times\sum_{l'=0}^{l_e-l_a}\exp\,(-i2\pi l'z/P)\,.\,F^0_{h'k'l'}\,, \tag{7a}$$

$$F''_{h'yz}=\exp\,(-i2\pi k_ay/O)\times\sum_{k'=0}^{k_e-k_a}\exp\,(-i2\pi k'y/O)\,.\,F_{h'k'z}\,, \tag{7b}$$

and:

$$F'''_{xyz}=\exp\,(-i2\pi h_ax/N)\times\sum_{h'=0}^{h_e-h_a}\exp\,(-i2\pi h'x/N)\,.\,F''_{h'yz}\,. \tag{7c}$$

The last transformation yields the desired result:

$$\varrho(xyz)=F'''_{xyz}\,. \tag{8}$$

Equations (7) are of the same form. It is remarkable that they do not represent simple FT's: each of the complex numbers in a transformed row remains to be multiplied by an additional phase factor dependent on its running index. This feature has to be considered in the development of a generally applicable procedure for the FFT of crystallographic data.

The equations (6) to (8) suggest a procedure involving four discrete steps: (1) The entire space of Fourier coefficients $F^0_{h'k'l'}$ is generated. The position of every point in this space is indicated by the transformed indices h', k', and l'. (2) Rows of coefficients $F^0_{h'k'l'}$ with common indices h' and k' are written into an auxiliary linear array, one at a time. The array is transformed, and one by one its elements are multiplied by the appropriate additional phase factor. As these factors depend only on the index of the transformed element, they are stored for the whole duration of the second step. The transformed elements $F'_{h'k'z}$ replace the untransformed ones, which are no longer needed. (3) Each row of Fourier coefficients $F'_{h'k'z}$ with equal indices h' and z is transformed. The $F''_{h'yz}$ take the storage place of the $F'_{h'k'z}$. (4) The $F''_{h'yz}$ with common indices y and z are treated in the same manner. Their place is taken by F'''_{xyz}. The final result can be output on a file.

Slant planes

Let

$$h_1x+k_1y+l_1z=p_1 \tag{9}$$

be the equation of a slant plane laid through the oblique coordinate system of a crystal. The transform at points (y,z) in this plane is obtained by the combination of equations (1) and (9) (Rollett, 1965):

$$\begin{aligned}\varrho(yz)_{\text{plane}}=\frac{1}{V}\sum_{h=h_a}^{h_e}\exp\,[-i2\pi h(p_1-k_1y-l_1z)/(h_1N)]\\ \times\sum_{k=k_a}^{k_e}\sum_{l=l_a}^{l_e}F_{hkl}\,.\exp\,[-i2\pi(ky/O+lz/P)]\,.\end{aligned} \tag{10}$$

With the introduction of equations (4) and (7), this becomes:

$$\begin{aligned}\varrho(yz)_{\text{plane}}=\sum_{h'=0}^{h_e-h_a}\exp\,[-i2\pi(h'+h_a)\\ \times(p_1-k_1y-l_1z)/(h_1N)]\,.\,F''_{h'yz}\,.\end{aligned} \tag{11}$$

This equation provides a general tool for the calculation of slant planes within the FFT framework. Data are transformed twice by means of the standard procedure outlined above, [equations (6–7*b*)]. Only the last step has to be done separately for each desired point (y,z).

The use of external storage devices

The need of the entire space of Fourier coefficients results in rather high storage requirements. For instance, at a given minimum resolution of 0·5 Å and cell constants of *ca.* 12 Å, each axis has to be divided into intervals of $\frac{1}{32}$. Simultaneous storing of the real and imaginary parts of the Fourier coefficients leads to a requirement of $2 \times 32 \times 32 \times 32 = 65536$ allocations.

Keeping this large amount of information in the rapid-access storage for the entire time of the calculations would be extremely wasteful and in many cases even impossible. However, an attempt to decrease by symmetry arguments the quantity of numbers to be stored would lead to difficulties in principle, as the FFT algorithm can transform whole periods only. In this work, external storage has been successfully used for a general implementation of the procedure outlined above.

Transportation to and from external media, however, is a time-consuming process compared to the calculations done in core. Even with modern computers, where interrupt techniques reduce the core time for a transport to the bare access time, extensive use of the external storage can make these steps time-determining for the whole algorithm. Furthermore, the waiting time for output may become unreasonably long.

For the following more detailed discussion, let us assume a right-handed coordinate system. In the very beginning, the space defined is occupied by Fourier coefficients indexed as $F_{h'k'l'}$. By successive transformations in each row and column they are finally replaced by the F'''_{xyz}. Diagrams of this final result are printed out for equal heights of z; and are therefore parallel to the crystal axes **a** and **b**. Different setups can generally be achieved by cyclic permutation of the indices h', k', and l' involved.

In the transformation procedures, rows of points with two indices in common are treated. This feature suggests transporting them as records to and from the external storage. The indexing of these records is linear. For one of the transformations however non-neighbouring records are required. Thus, the storage device should have random-access features (*e.g.*, drum or disc). In this case, the preliminary filling with Fourier coefficients can be done effectively without any preliminary sorting of the data.

At first glance, every Fourier coefficient appears to have to be transported to and from the external storage device for each of the three transformations. However, one plane of Fourier coefficients in the example quoted above would comprise only $2 \times 32 \times 32 = 2048$ numbers. Arrays of this size can easily be handled in computers of medium size, thus allowing for two transformations on one transport.

The sequence of the transformations is arbitrary; thus, the direction of the records in the outlined coordinate system can be chosen freely. This choice and the sequence of the three transformations, however, contribute very much to the effectiveness of the method.

For instance, let us assume the $l'(z,c)$ axis as the direction of the records: obviously, the first two transformations would have to be done in xz or yz planes. In order to obtain one of the finally desired xy planes, all of the records would have to be input from disc or drum each time.

This disadvantage is avoided when the records are chosen parallel to one of the axes in the plane later to be printed out. Then the second transformation is done in the direction of the $l'(z,c)$ axis. After the last transformation, the xy planes can be printed without any further transportation to or from the auxiliary storage.

Time requirements

With the neglect of additions, the computer time for the three transformation steps is given by:*

1. Transformation in x direction:

$$t_x = O\,.\,P\,.\,t_{\text{get}}(2N) + O\,.\,P\,.\,(t_{\text{FFT}}(N) + t_{\text{mul}}\,.\,N)\,, \quad (12a)$$

2. Transformation in z direction:

$$t_z = N\,.\,O\,.\,(t_{\text{FFT}}(P) + t_{\text{mul}}\,.\,P) + O\,.\,P\,.\,t_{\text{put}}(2N)\,, \quad (12b)$$

and

3. Transformation in y direction:

$$t_y = O\,.\,P\,.\,t_{\text{get}}(2N) + N\,.\,P\,.\,(t_{\text{FFT}}(O) + t_{\text{mul}}\,.\,O)\,. \quad (12c)$$

Rough time values for the Telefunken TR440 computer are:

(*a*) time to put a record on disc:
$t_{\text{put}} \simeq 4$ msec (for 256 words, linearly mounting);
(*b*) time to get a record from disc:
$t_{\text{get}} \simeq 18$ msec (for 256 words, linearly mounting);
(*c*) time for complex multiplication:
$t_{\text{mul}} \simeq 0{\cdot}08$ msec;
(*d*) time for FFT (library procedure *FOUCOM*, Telefunken, 1971)
$t_{\text{FFT}} = 2{\cdot}88 + 49{\cdot}7N \log_2 N/1000$ msec.

The transformation of a space with $32 \times 32 \times 32$ complex data points thus would be estimated to take about 80 sec. Since the time required is independent of the quantity of the data, the method can be expected to be superior to conventional ones when Fourier transforms of medium to large data sets are to be calculated.

An implementation should take care of some highly redundant features that the described algorithm has to this point: 1. In the first step, the majority of transformations would have to be carried out on zeros. For instance, while the space of coefficients in the example comprises about 32000 complex data points, a medium-sized data set of 1000 independent reflexions in an orthorhombic space group would require only 25% of the allotted space. With all input data real, this percentage would drop to 13%. 2. In addition, the result obtained is for an entire unit cell. Generally, however, only an asymmetric unit is required. Points

* The calculation assumes that a record stores both the real and imaginary parts of a Fourier coefficient.

of interest in the rest of the unit cell are easily generated by application of the symmetry operations to the corresponding points of the asymmetric unit.

ALGOL program

A computer-orientated program has been written implementing an improved version of the described algorithm for the FFT of three-dimensional crystal data in all space groups (Lange, Stolle & Huttner, 1972). ALGOL 60 (Baumann, 1968) was chosen as the coding language because of its high documentational value. Only the simpler elements of this language have been used, which are also available in FORTRAN.

An assembler procedure has been used for the FFT steps themselves (Telefunken, 1971). It can be replaced by any algorithm which calculates one-dimensional Fourier transforms; radix-2 algorithms (Barth & Kremer, 1969; Markel, 1971) can be adopted as well as mixed-radix versions (Singleton, 1969). Trigonometric functions for phase factors are drawn from cosine tables containing an entire period.

Real and imaginary parts of the Fourier coefficients are handled in one record throughout the program. A Boolean array notes whether a record exists on the background storage or not. Only rows containing at least one non-zero element have to be kept and transformed.

In the preliminary stage of the process, all the space of Fourier coefficients has to be generated. Indices and magnitudes of the dependent Fourier coefficients are readily calculated by Waser's method (Waser, 1955; Wells, 1965), which uses symmetry relationships. In order to sort the indexed information properly into records, maximum use is made of the rapid access storage that is dynamically achievable in ALGOL 60. A working array is filled by the images of as many records as possible. With all coefficients real (centrosymmetric case) the imaginary parts are not stored. Whenever the working array is filled, the generated records are written into the external storage device to adresses kept in a special list.

The direction of the records was chosen parallel to the **a** axis. The records are transformed first; the second and the third transformations are done in the directions of the **c** and **b** axes respectively. Transformation sequences that differ from this pattern are achieved by a cyclic permutation of the transformed indices h', k' and l' involved.

With all real data, about one half of the transformation time of the first step is saved by the simultaneous FFT of two rows (Cooley, Lewis & Welch, 1969*b*). Only the desired rows are transformed for the third time. The sequence of their transformation is given by the form of output. Special pointer tables keep the necessary information for rapid access.

The option of calculating slant planes has been implemented in the program for the FFT of crystallographic data. Extensive use is made of pointer tables. Whenever records are transported into core storage for the third transformation, points of the slant planes are calculated and stored externally. Completion and output is done after the finish of all third transformations.

Table 1 gives some time values for the Fourier transformation of a medium-sized data set on a TR440 computer. For a given resolution there is a drop in computer time by a factor of about 8.

Table 1. *Example for the* FFT *of a crystallographic set of data*

Literature: Huttner & Lange (1972)
Space group: $P2_1/c$
Cell constants: 15·5, 8·7, 16·7 Å, $\beta=90°$
Number of data: 2696
Resolution: ≤0·3 Å
Limits of summation: $x=0\ldots0·5$, $y=0..1$, $z=0..0·5$
Z axis chosen as **c**

Beevers–Lipson: total transformation time: ~17 min*

FFT: resulting space of Fourier coefficients: 64 × 32 × 64 points
theoretical number of records: 32 × 64

Time values:
(*a*) filling disc
number of non-zero records: 609; time needed: 26·0 sec.
(*b*) first transform
length of records: 64;
records transformed: 609 (29·73 %); time needed: 18·4 sec.
(*c*) second transform
length of columns: 64
records transformed: 1344 (65·63 %); time needed: 52·1 sec.
(*d*) third transform
length of rows: 32;
rows transformed: 1089 (26·59 %); time needed :30·9 sec;
time for total transformation: 133:2 sec.

* The Fourier program used was a modified local version of an algorithm in ALGOL developed at the University of Manchester.

The authors are indebted to the Deutsche Forschungsgemeinschaft for financial support. Computing facilities were kindly provided for by the Leibniz Rechenzentrum der Bayerischen Akademie der Wissenschaften.

References

Barth, W. & Kremer, H. (1969). *Elektron. Datenverarb.* **12**, 570–573.

Baumann, R. (1968). *ALGOL-Manual der ALCOR-Gruppe.* München: Oldenburg.

Bondot, P. (1971). *Acta Cryst.* A**27**, 492–494.

Brenner, N. M. (1969). *IEEE Trans.* AU-**17**, No. 2, 128–132.

Cooley, J. W. & Tukey, J. W. (1965). *Math. Comput.* **19**, 297–301.

Cooley, J. W., Lewis, P. A. W. & Welch, P. D. (1969*a*). *IEEE Trans. Educ.* **12**, No. 1, 27–34.

Cooley, J. W., Lewis, P. A. W. & Welch, P. D. (1969*b*). *IEEE Trans.* AU-**17**, No. 2, 77–85.

Gentleman, W. M. & Sande, G. (1966). *AFIPS Conf. Proc.* **29**, 563–578.

HOPPE, W., GASSMANN, J. & ZECHMEISTER, K. (1970). In *Crystallographic Computing*, pp. 22–36. Edited by F. R. AHMED. Copenhagen: Munksgaard.
HUTTNER, G. & LANGE, S. (1972). *Acta Cryst.* B**28**, 2049–2060.
IEEE Transactions (1967). AU-**15**, No. 2, Special issue on the FFT.
IEEE Transactions (1969). AU-**17**, No. 2, Special issue on the FFT.
LANGE, S., STOLLE, U. & HUTTNER, G. (1972). *FFT3X-Ein Programm zur schnellen Fouriertransformation röntgenographischer Daten.* TU München. Copies and descriptions are available upon request.
MARKEL, J. D. (1971). *IEEE Trans.* AU-**19**, No. 4, 305–311.
ROLLETT, J. S. (1965). In *Computing Methods in Crystallography*, pp. 82–88. Edited by J. S. ROLLETT. Oxford: Pergamon Press.
SINGLETON, R. C. (1968*a*). *Commun. Assoc. Comput. Mach.* **11**, No. 11, 773–776.
SINGLETON, R. C. (1968*b*). *Commun. Assoc. Comput. Mach.* **11**, No. 11, 776–779.
SINGLETON, R. C. (1969). *IEEE Trans.* AU-**17**, No. 2, 93–103.
STOUT, G. H. & JENSEN, L. H. (1968), p. 246. *X-ray Structure Determination.* New York: Macmillan.
TELEFUNKEN (1971). *TR440-Programmbeschreibung ALGOL. Konstanz.* Beschr. Nr. N 31. F2. 11, S. 4–5.
WASER, J. (1955). *Acta Cryst.* **8**, 595.
WELLS, M. (1965). *Acta Cryst.* **19**, 173–179.

Acta Cryst. (1973). A**29**, 449

The Theory of the Spin-Density Patterson Function

BY C. WILKINSON

Queen Elizabeth College, University of London, Campden Hill Road, London W8 7AH, England

(*Received* 18 *December* 1972; *accepted* 11 *February* 1973)

A general treatment is given of the derivation of a spin-density Patterson function from unpolarized neutron-diffraction data, showing that a peak in the function due to atoms m and n with spins $\mathbf{S}^m$, $\mathbf{S}^n$ has a height proportional to $\mathbf{S}^m \cdot \mathbf{S}^n$ and is elongated in a direction which bisects these spin directions. The elongation is not appreciably affected even when the spin-density distributions of the atoms are highly aspherical.

Introduction

The essential features of the spin-density Patterson function have been described in a previous paper (Wilkinson, 1968). The purpose of the present paper is to provide a more precise treatment of certain parts of theory in which approximations were previously made. The same notation will again be used.

The Patterson function $Q'(u)$ calculated from unpolarized neutron-diffraction intensities from a single crystal is given by

$$Q'(\mathbf{u}) = Q(\mathbf{u}) - R(\mathbf{u})$$

where $Q(\mathbf{u})$ is the scalar-product autocorrelation function of the spin density:

$$Q(\mathbf{u}) = \sum_{1-3}^{i} \sum_{1-3}^{j} \mathscr{S}_i(\mathbf{r}) * \mathscr{S}_j(-\mathbf{r}) \,.$$

$R(\mathbf{u})$ is the sum of the convolutions of the Fourier transform of the appropriate pairs of scattering-vector direction-cosine products with terms in the summation for $Q(\mathbf{u})$ and is

$$R(\mathbf{u}) = \sum_{1-3}^{i} \sum_{1-3}^{j} \overline{(k_i k_j)} * [\mathscr{S}_i(\mathbf{r}) * \mathscr{S}_j(-\mathbf{r})] \,.$$

In the previous theory for the evaluation of $R(\mathbf{u})$ it was assumed that the form factors for all magnetic atoms in the structure could be represented by the function $\exp(-p^2k^2)$. The present treatment shows that this is an unnecessary approximation, but that the main features of the result (*i.e.* Patterson peak height proportional to the scalar product of component spins, elongation of peak in direction bisecting spin directions) remain unchanged. Expressions for $R(\mathbf{u})$ and $Q(\mathbf{u})$ are derived for two and three-dimensional data in Appendices 1 and 2.

$Q'(\mathbf{u})$ for a zone of data measured out to $|\mathbf{k}| = k_0$

In Appendix 1 it is shown that for two-dimensional data the Patterson peak due to atoms m and n which are separated by a distance $\mathbf{u}^{mn}$ is

$$\begin{aligned} Q'(\mathbf{u}^{mn} + \mathbf{x}) = \operatorname*{Lt}_{k_0 \to \infty} \{ & \pi k_0^2 (\Lambda_1(2\pi r k_0) \\ & \times [S_\perp^m S_\perp^n + \tfrac{1}{2} S_\parallel^m S_\parallel^n \cos(\omega_m - \omega_n) \\ & - S_\parallel^m S_\parallel^n \cos(2\alpha - (\omega_m + \omega_n))] \\ & + \Omega(2\pi r k_0) S_\parallel^m S_\parallel^n \cos(2\alpha - (\omega_m + \omega_n)) \\ & * (\hat{\mathscr{S}}^m(\mathbf{r}) * \hat{\mathscr{S}}^n(-\mathbf{r})) \} \,. \end{aligned}$$

The component functions in this expression require some explanation.

(i) The quantity $\hat{\mathscr{S}}^m(\mathbf{r}) * \hat{\mathscr{S}}^n(-\mathbf{r})$ is similar to that which appears in the expression for Patterson peaks calculated from X-ray intensities. In the two-dimensional case it represents the convolution of the pro-