COWLEY, J. M. & IIJIMA, S. (1972). *Z. Naturforsch.* **27**a, 445–451.

GATEHOUSE, B. M. & WADSLEY, A. D. (1964). *Acta Cryst.* **17**, 1545–1554.

GRUEHN, R. & NORIN, R. (1967). *Z. anorg. allgem. Chem.* **355**, 176–181.

GRUEHN, R. & NORIN, R. (1969). *Z. anorg. allgem. Chem.* **367**, 209–218.

IIJIMA, S. (1973). *Acta Cryst.* A**29**, 18–24.

IIJIMA, S. & ALLPRESS, J. G. (1973). *J. Solid State Chem.* **7**, 94–105.

IIJIMA, S. & ALLPRESS, J. G. (1974*a*). *Acta Cryst.* A**30**, 22–29.

IIJIMA, S. & ALLPRESS, J. G. (1974*b*). *Acta Cryst.* A**30**, 29–36.

IIJIMA, S., KIMURA, S. & GOTO, M. (1973). *Acta Cryst.* A**29**, 632–636.

KIMURA, S. (1973). *J. Solid State Chem.* **6**, 438–449.

NIMMO, K. M. & ANDERSON, J. S. (1972). *J. Chem. Soc. Dalton*, pp. 2328–2337.

NORIN, R. (1963). *Acta Chem. Scand.* **17**, 1391–1404.

NORIN, R. (1965). *Acta Chem. Scand.* **20**, 871–880.

ROTH, R. S. & WADSLEY, A. D. (1965). *Acta Cryst.* **18**, 724–730.

SCHÄFER, H., BERGNER, D. & GRUEHN, R. (1969). *Z. anorg. allgem. Chem.* **365**, 31–50.

WADSLEY, A. D. & ANDERSSON, S. (1970). *Perspectives in Structural Chemistry*, Edited by J. D. DUNITZ and J. A. IBERS, Vol. 3. New York: John Wiley.

Acta Cryst. (1974). A**30**, 257

The Use of Symmetry with the Fast Fourier Algorithm

BY DAVID A. BANTZ AND MARTIN ZWICK

Department of Biophysics, The University of Chicago, Chicago, Illinois 60637, *U.S.A.*

(*Received* 9 *August* 1973; *accepted* 14 *September* 1973)

This paper presents an algorithm for making use of symmetry in the fast Fourier transform in a simple and general way which is applicable to nearly all space groups. This allows one to reduce storage requirements to approximately what is needed for an asymmetric unit of the electron-density function, and hence makes possible economical forward and reverse transforms of large unit cells in core.

Introduction

In recent years the 'Fast Fourier Transform' (FFT) of Cooley & Tukey (1965) has been increasingly applied to problems in crystallography and electron microscopy. A consideration limiting its use, however, has been the fact that, while Friedel symmetry may be conveniently incorporated into the algorithm, it has only recently been possible to make use of space-group symmetry to reduce storage and computing time requirements. The storage problem is more serious since it frequently happens, particularly for crystals of large biological molecules, that the available core storage is not sufficient to include the entire unit cell, which is what the fast Fourier algorithm normally requires. Exceeding the core limitations forces one to use a more complex and time-consuming form of the algorithm which uses tape or disk for storage (Gentleman & Sande, 1966; Brenner, 1968, 1972; Singleton, 1968; Hubbard & Quicksall, 1970). The net result of not using space-group symmetry and having to include the entire unit cell in the transform is to erode the savings possible with the FFT; indeed, for high-symmetry space groups, the fast Fourier method may not have a significant advantage over conventional algorithms, and may even be costlier. For these reasons, an approach which allows the FFT to make use of space-group symmetry is of some value, as this could provide the storage factor necessary to allow the computations to be done in-core.

We present such an approach. A method which bears some similarities to the procedures we set out here, has been proposed by Ten Eyck (1973), who has analyzed how the fast Fourier algorithm might be modified to include various possible symmetries. The present approach differs from Ten Eyck's in being simpler and more general since only standard, unmodified, fast Fourier subroutines are used. On the other hand, it suffers from the disadvantage of making full use of symmetry only for storage, but not for time reductions.

Procedure

For space-group symmetry with N general positions given by rotations and translations, S^j and $\mathbf{t}^j$, $j=1, \ldots, N$, the electron-density function and its transform have the symmetry,

$$\varrho(S^j\mathbf{x}+\mathbf{t}^j)=\varrho(\mathbf{x}) \tag{1}$$

and

$$F(\tilde{S}^j\mathbf{h})=\exp\,(-2\pi i\mathbf{h}\,.\,\mathbf{t}^j)F(\mathbf{h}) \tag{2}$$

since

$$\tilde{S}^j\mathbf{h}\,.\,\mathbf{x}=\mathbf{h}\,.\,S^j\mathbf{x}=\mathbf{h}\,.\,(\mathbf{x}^j-\mathbf{t}^j)\,. \tag{3}$$

Centering is included in these equations by operations of the form $[S^j,\mathbf{t}^j+\boldsymbol{\tau}]$ where $\boldsymbol{\tau}$ is the appropriate translation. For example, in *C*-centered cells, general positions occur in pairs: $[S^J\mathbf{x}+\mathbf{t}^J, S^J\mathbf{x}+\mathbf{t}^J+\boldsymbol{\tau}]$ where $\boldsymbol{\tau}=+(\frac{1}{2},\frac{1}{2},0)$ and $j=1\ldots N/2$. Note that this gives, *via* equation (2), the expected systematic absences. A related treatment of these symmetry equations has been given by Bienenstock & Ewald (1962).

We can use (1) and (2) to generate a complete set of data from an asymmetric unit (a.u.) in real or reciprocal space,† *i.e.* we need roughly $1/N$ of the complete set (slightly larger because the boundary of the a.u. must be included). We can make use of the symmetry in the transform by calculating, for each l,

$$G_l(x,y)=\sum_h\sum_k F_l(h,k)\,[\exp 2\pi i(hx+ky)]\,, \qquad (4)$$

where h and k range over the whole hk plane, and the two-dimensional $F_l(h,k)$ are generated from a reference array of the a.u. $F(h,k,l)$ by equation (2). The electron-density function is then obtained by a series of one-dimensional transforms on $G_l(x,y)$:

$$\varrho_{xy}(z)=\sum_l G_{xy}(l)\exp(2\pi ilz)\,. \qquad (5)$$

This mixed function, G, we write variously as $G_l(x,y)$ and $G_{x,y}(l)$ to emphasize that the calculation takes place in two stages: a series of two-dimensional Fourier syntheses in which we transform between real space (x,y) and reciprocal space (h,k), and a series of one-dimensional transforms, in which (x,y) are held fixed and we transform on z or l. These one-dimensional transforms need only be done for (x,y) in the asymmetric unit of real space, and hence the $G_l(x,y)$ need only be saved for these values of (x,y). This very simple fact is the basis for the present approach.

Let us consider the storage requirements for ϱ, F, and G. The asymmetric unit in ϱ requires about $n_x\,.\,n_y\,.\,n_z/N$ real storage elements where $1/n_x$ is the sampling interval along the x axis, and similarly for n_y and n_z. The domain in reciprocal space which can be used to generate ϱ is, like the real-space cell, defined by a parallelepiped of $n_x\,.\,n_y\,.\,n_z$ points, and thus the asymmetric unit for the F's is again about $n_x\,.\,n_y\,.\,n_z/N$ points. If the F's are real, this is the number of standard-length storage elements needed; if they are complex, Friedel symmetry reduces the number of points by $\frac{1}{2}$, and gives the same requirement for storage.

We now consider the storage requirements for $G(x,y,l)$. From equation (5), it is clear, from the reality of ϱ, that

$$G_{xy}(-l)=G^*_{xy}(l)\,. \qquad (6)$$

This means that equation (4) need be evaluated only for $l\geq 0$.

Equation (4) gives us, for any l, values of $G_l(x,y)$ for the full plane of $n_x\,.\,n_y$ points, but because of the space-group symmetry, we need save only a fraction, $n_x\,.\,n_y/M$ $(M\leq N)$ of these values, for use in the subsequent one-dimensional Fourier of equation (5). The same domain of G in xy is saved, for each l, thus reducing, at this point, the total storage requirements for $G(x,y,l)$ to $n_x\,.\,n_y\,.\,n_z/M$. In addition mirror planes normal to z will result in G being real (if the mirror is at $z=0$) or of known phase, and yield a further reduction in storage requirements by a factor of two.‡ We shall show that in most cases M or $2M$ is equal to N. More specifically, for all space groups not rhombohedral or cubic, all N general positions can be used to reduce the storage requirements of G to $n_x\,.\,n_y\,.\,n_z/N$ values, *i.e.* the same number needed for ϱ and F.

This results from the fact that, except for the two cited systems, $S^J(1,3)=S^J(2,3)=S^J(3,1)=S^J(3,2)=0$ and $S^J(3,3)=\pm 1$ for all j; *i.e.* $[S^J,\mathbf{t}^J]$ are given by

$$S^J=\begin{bmatrix} a & b & 0\\ c & d & 0\\ 0 & 0 & \pm 1\end{bmatrix} \quad \mathbf{t}^J=\begin{bmatrix} t^j_x\\ t^j_y\\ t^j_z\end{bmatrix}. \qquad (7)$$

From equation (1), all $[S^J,\mathbf{t}^J]$ operations of the above form which differ in one or more of the a, b, c, d, t_x, t_y variables can be used to reduce the domain of x, y for which $G_l(x,y)$ must be saved and the Fourier transform ofequation (5) calculated.§ $[S^J,\mathbf{t}^J]$ operations differing only in the sign of $S(3,3)$ and (perhaps) t_z represent mirror planes normal to z; in this case, $G_l(x,y)$ has known phase and hence we need save only the magnitude of G (see footnote ‡). The net effect of all this is that the total storage for G will be approximately the same as that needed for the asymmetric unit of F or ϱ.

The computational procedure used is outlined schematically in Fig. 1. The asymmetric unit of F is stored in planes of l, for $l\geq 0$. For each l, equation (2) is used to generate a complete $F_l(h,k)$ plane in a two-dimensional working array (*a*). The two-dimensional transform of this function is taken in place to give $G_l(x,y)$ for the complete x,y plane. Only the portion of this G array needed for the calculation of ϱ need be retained. Since the values of the asymmetric unit $F_l(h,k)$ are required only for the calculation of $G_l(x,y)$, the portion of the G array to be retained may be stored in the same locations as the a.u. $F_l(h,k)$. When this procedure is done for all l, and the asymmetric unit F's fully replaced by G's, $G_{xy}(l)$ values are collected for each (x,y) in the asymmetric unit and the one-dimensional transform of equation (5) is performed (*b*). A working array is used for this calculation, but the resulting $\varrho_{xy}(z)$

† Naturally we also make use of the reality of ϱ, and the consequent Hermitian symmetry of F.

‡ If the mirror is at $z=m$, let $\varrho'(z)=\varrho(z+m)$ and $G'(l)$ be the transform of ϱ'. Then ϱ' is centric, G' is real and $G_{xy}(l)=\exp[-2\pi ilm].\,G'(l)$. Thus we need save only the (signed) magnitude of $G'(l)$ rather than a complex value. The phase of G is given by the equation just cited; this is what we mean by 'known phase'.

§ If hexagonal axes are used for the rhombohedral space groups all $[S^J,\mathbf{t}^J]$ will be of the form given in equation (7). For cubic space groups, one third of the general positions are of this type and can be used for storage reduction.

values are again stored in place of the corresponding $G_{xy}(l)$ array. Thus, the same storage area is used for the asymmetric unit of F, for the reduced G array, and for the final asymmetric unit of ϱ.

In the description given here (and in the program to be described below) we make use of, for the small one- and two-dimensional transforms, a perfectly general fast Fourier routine, incorporating at most Friedel symmetry. While this procedure is then immediately applicable to any space-group symmetry, simply by specifying the set of $[S^J, \mathbf{t}^J]$, it would be possible to use more complicated fast Fourier algorithms which make intrinsic use of some symmetry, as described by Ten Eyck (1973).

The above procedure is readily 'reversed' so that we easily calculate the inverse transform, $\varrho(xyz) \rightarrow F(hkl)$ by series of one-dimensional (inverse) transforms on z to yield $G_{xy}(l)$ followed by two-dimensional (inverse) transforms. That is, we simply invert equations (5) and (4). Again, we need only store, at any one time, an asymmetric unit of data. The ability to transform back and forth economically between real and reciprocal space makes practical some new approaches to direct methods [see, for example, Barrett & Zwick (1971)].

While full use is made of the space-group symmetry to reduce storage requirements (except for the cubic system), the time needed for the transform is only slightly less than that of an $n_x . n_y . n_z$ point in-core fast Fourier transform. The calculation consists of $(n_z/2+1)$ two-dimensional transforms of size $n_x . n_y$ [equation (4)] followed by $n_x . n_y/M$ one-dimensional transforms of size n_z, and thus time savings are only achieved in the second stage of the calculation. If, following Ten Eyck, we used two-dimensional symmetry in the two-dimensional transform steps, further time savings would be possible. The advantage of the present approach is the generality of this algorithm for all space groups.

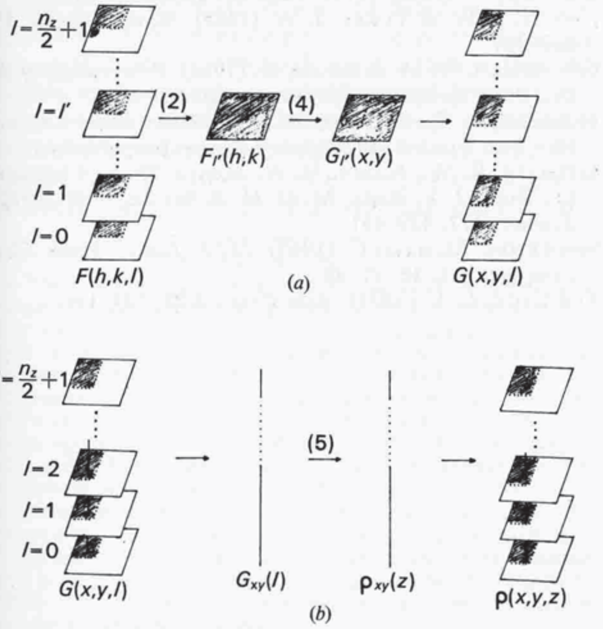

Fig. 1. Schematic representation of transform from $F(h,k,l)$ to $\varrho(x,y,z)$. Shaded portions of arrays indicate the portion actually stored; numbers in parentheses indicate equations in the text. (a) Transform, plane by plane, to obtain $G_l(x,y)$. (b) One-dimensional transforms from $G_l(x,y)$ to $\varrho(x,y,z)$. Asymmetric units in xy or hk need not actually be parallelograms.

To estimate the time required for this calculation, assume $n_x = n_y = n_z = n = 2^k$. We assume a routine which utilizes Friedel symmetry is used. The time for a full n^3 point transform with the standard fast Fourier procedure is proportional to $n^3 \log_2 n^3 = 2^{3k} . 3k$ (Cooley & Tukey, 1965). The above algorithm requires a time proportional to $2(n/2+1)n^2 \log_2 n^2 \simeq 2^{3k}2k$ for the two-dimensional transforms (where the additional factor of two results from the absence of Friedel symmetry in these transforms) plus $n^2 . n \log_2 n/M = 2^{3k}k/M$ for the one-dimensional transforms which do have Friedel symmetry. The ratio of these two times, which is the time reduction factor relative to the non-symmetry-utilizing FFT, is

$$\frac{2^{3k}(2k+k/M)}{2^{3k}(3k)}$$

and clearly cannot be less than $\frac{2}{3}$. The time savings relative to conventional Fourier algorithms are, of course, much greater. In fact, the most time-consuming portion of calculating electron-density maps with the fast Fourier is often printing the map in suitable format.

A program, called *CHAFF*, implementing this algorithm, space-group independent and for arbitrary dimensions (not restricted to powers of 2) has been written in FORTRAN. It takes transforms in either direction and while only the asymmetric unit of F or ϱ is stored, the user can easily access (for input or output) any symmetry-related (h,k,l) or (x,y,z). Thus, although the data is stored in an asymmetric unit chosen for convenience in the calculation ($0 \leq z < 1$ or $0 \leq z < \frac{1}{2}$ for centric space groups and a domain in xy approximately $1/M$ of the plane), for purposes of input and output or modifying values, the user can use any definition of asymmetric unit he or she chooses.

This program is currently being used for Patterson and Fourier calculations on yeast $\mathrm{tRNA}^{\mathrm{met}}_{\mathrm{F}}$, space group $P6_422$ (Schevitz, Navia, Bantz, Cornick, Rosa, Rosa & Sigler, 1972). Actual storage requirements and running times on an IBM-360/195 are given in Table 1. For comparison, figures are also shown for the in-core non-symmetry-utilizing fast Fourier based on a program of N. Brenner and a disk version of this program which we have also implemented. The core requirements include subroutines for output of maps, but the times shown do not include time for this output.

For many applications, the algorithm described here

Table 1. *Storage and time requirements for different fast Fourier algorithms* (IBM 360/195)

Times for taking a transform and total core requirements using three fast-Fourier programs. *CHAFF* is the symmetry-utilizing in-core program described in this paper. Since *IN-CORE* and *DISK* do not use symmetry, transform times and core requirements are space-group independent. Numbers in parentheses are extrapolated. The times listed include the time for initialization and data input before transform. This is roughly proportional to the number of reflections.

		CHAF		*IN-CORE*		*DISK*	
Space group	Dimensions	Time (sec)	Core* (bytes)	Time‡ (sec)	Core* (bytes)	Time (sec)	Core* (bytes)
$P6_222$	32·32·32	1·6	148k	1·7	222k	11·7	138k
	32·32·33	1·8	148k	3·3	352k	–†	–†
	64·64·72	6·7	288k	(25)	(2392k)	–†	–†
$P2_12_12_1$	16·16·32	1·2	134k	1·3	122k	4·3	114k
	32·32·32	2·2	174k	2·2	222k	11·7	138k
	64·64·32	5·0	328k	4·5	614k	89·1	234k
	64·64·64	7·4	464k	(7)	(1142k)	(160)	(234k)
$I422$	32·32·32	1·7	156k	1·4	222k	10·8	138k
	64·64·64	6·5	258k	(7)	(1142k)	(160)	(234k)
	64·64·96	8·8	296k	(40)	(3128k)	–†	–†

* Numbers include buffer space which can be reduced slightly with some loss of efficiency. A 'k' of storage is 1024 bytes; a byte is an 8-bit unit of storage. Maximum storage available varies from one installation to another, but at present rarely exceeds about 1000k .

† *DISK* program requires powers of 2.

‡ The *IN-CORE* program uses a less efficient Fourier routine if any dimension is not a power of 2. It could be modified to reduce time to approximately that of *CHAFF.*

will allow Fourier calculations to be done without the use of external storage, or, if the entire unit cell will fit into available core, with far less storage than otherwise required. For very large unit cells, however, even an asymmetric unit may be too large to be accommodated in core. For such cases the *CHAFF* algorithm could be modified to utilize external storage (this has not actually been implemented) and since only an asymmetric unit of data is needed, input–output costs would be approximately $1/M$ of that required for the non-symmetry-utilizing program (called *DISK* in Table 1).

This research was supported by grants from the U.S. Public Health Service (GM 15225), National Science Foundation (GB 27645 and 27667) and The American Cancer Society (Np98A).

References

Barrett, A. N. & Zwick, M. (1971). *Acta Cryst.* A**27**, 6–11.

Bienenstock, A. & Ewald, P. P. (1962). *Acta Cryst.* **15**, 1253–1261.

Brenner, N. (1968, 1972). Personal communications.

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

Gentleman, W. M. & Sande, G. (1966). *Proceedings of the Fall Joint Computer Conference*, pp. 563–578.

Hubbard, C. R. & Quicksall, C. (1970). 'Ames Laboratory Fast Fourier' Preliminary Report (unpublished).

Schevitz, R. W., Navia, M. A., Bantz, D. A., Cornick, G., Rosa, J. J., Rosa, M. D. H. & Sigler, P. B. (1972). *Science*, **177**, 429–431.

Singleton, Richard C. (1967). *IEEE Trans. Audio Electroacoust.* **AU-15**, 91–98.

Ten Eyck, L. F. (1973). *Acta Cryst.* A**29**, 183–191.