Acta Crystallographica Section A
Foundations of Crystallography
ISSN 0108-7673

Received 23 July 1999
Accepted 10 January 2000

Discrete Fourier transform in arbitrary dimensions by a generalized Beevers–Lipson algorithm

Martin Schneider and Sander van Smaalen*

Laboratory of Crystallography, University of Bayreuth, D-95440 Bayreuth, Germany. Correspondence e-mail: smash@uni-bayreuth.de

The Beevers–Lipson procedure was developed as an economical evaluation of Fourier maps in two- and three-dimensional space. Straightforward generalization of this procedure towards a transformation in n-dimensional space would lead to n nested loops over the n coordinates, respectively, and different computer code is required for each dimension. An algorithm is proposed based on the generalization of the Beevers–Lipson procedure towards transforms in n-dimensional space that contains the dimension as a variable and that results in a single piece of computer code for arbitrary dimensions. The computational complexity is found to scale as $N\log(N)$, where N is the number of pixels in the map, and it is independent of the dimension of the transform. This procedure will find applications in the evaluation of Fourier maps of quasicrystals and other aperiodic crystals, and in the maximum-entropy method for aperiodic crystals.

1. Introduction

Numerical evaluation of Fourier transforms is an important tool in crystallography. It includes the computation of various types of Fourier map from experimental diffraction data and the generation of a set of calculated structure factors from an electron-density function as part of the maximum-entropy method.

For crystals with three-dimensional (3D) translational symmetry, the transform is three-dimensional. Aperiodic crystals lack 3D translational symmetry. The structures of all types of aperiodic crystal (quasicrystals, incommensurately modulated structures and incommensurate composite crystals) can be characterized by a periodic density function in n-dimensional (nD) space with $n \geq 4$ (van Smaalen, 1995). The crystallographic study of aperiodic crystals thus requires Fourier transforms in nD space (Dusek & Petricek, 1998).

An efficient algorithm for a multidimensional Fourier transform is provided by the Beevers–Lipson method when it is combined with a 1D fast Fourier transform (FFT) (Immirzi, 1976). Originally, it was developed for the computation of the electron density from the structure factors of 2D and 3D periodic crystals. The usual implementation explicitly employs the 3D nature of the transform (Bricogne, 1996). A straightforward generalization towards more than three dimensions is possible, but it requires seperate computer code for each dimension.

For nD crystallography, a computer program is only available for transforms up to $n = 6$ (Dusek & Petricek, 1998). More than six dimensions are required for several modulated crystals [*e.g.* α-U between 38 and 43 K (Walker, 1986; van Smaalen & George, 1987) and $(TaSe_4)_2I$ in its multi-dimensional charge-density-wave state (Lorenzo *et al.*, 1993) require $n = 7$] and for the modulated icosahedral quasicrystal AlCuFe ($n = 12$) (Perez-Mato & Elcoro, 1994).

In this contribution, we describe a generalization of the Beevers–Lipson algorithm towards more than three dimensions, which results in a single piece of computer code that can handle transforms of an arbitrary number of dimensions.

2. The *n*-dimensional Fourier transform

The problem is considered of the discrete Fourier transform in nD space. To this end, the continuous electron density in the nD unit cell is discretized on a symmetry-adapted grid of

$$N_{\rm pix} = N_1 \times \ldots \times N_n \qquad (1)$$

pixels (Fig. 1). The origin of the unit cell is chosen to coincide with the pixel $(0, \ldots, 0)$. The coordinates of the pixels are taken along the basis vectors of the unit cell and the shape of each voxel is that of the Wigner–Seitz cell of the direct lattice in nD space.

The structure factor of an aperiodic crystal ($n > 3$) or a periodic crystal ($n = 3$) is defined as the Fourier transform of the electron density of one nD unit cell (Janssen *et al.*, 1992; van Smaalen, 1995):

$$F(\mathbf{H}) = \int_0^1 {\rm d}x_1 \ldots \int_0^1 {\rm d}x_n \, \rho_s(\mathbf{x}_s) \exp[2\pi i \mathbf{H}\cdot\mathbf{x}_s], \qquad (2)$$

where the scattering vectors in nD reciprocal space are defined by

$$\mathbf{H} = h_1\mathbf{a}_1^* + \ldots + h_n\mathbf{a}_n^*. \quad (3)$$

Position vectors in nD direct space are defined by their coordinates $(x_{s1}, \ldots, x_{sn})$ with respect to the basis vectors of the nD direct lattice:

$$\mathbf{x}_s = x_{s1}\mathbf{a}_1 + \ldots + x_{sn}\mathbf{a}_n. \quad (4)$$

$\rho_s(\mathbf{x}_s)$ is the generalized electron density in nD space and it is periodic in each of its n arguments. For $n = 3$, it is equal to the real electron density in the crystal. For $n \geq 4$, the electron density in real space follows as the 3D section of the generalized electron density, perpendicular to the $(n - 3)$ additional coordinates (van Smaalen, 1995).

The discretized electron-density function is defined by the values of $\rho_s(\mathbf{x}_s)$ on the grid of equation (1):

$$\rho_k = \rho_s(\mathbf{x}_s(k)), \quad (5)$$

with the coordinates of pixel $(i_1, \ldots, i_n)$ given by

$$\mathbf{x}_s(k) = (i_1/N_1, \ldots, i_n/N_n). \quad (6)$$

Each pixel is identified by a single number k, which is defined below. With these definitions, and expanding the exponential function, the discrete Fourier transform is obtained as (Bricogne, 1996)

$$\begin{aligned} F(\mathbf{H}) = &\sum_{i_n=0}^{N_n-1} \exp[2\pi i h_n i_n/N_n] \\ &\times \sum_{i_{n-1}=0}^{(N_{n-1})-1} \exp[2\pi i h_{n-1} i_{n-1}/N_{n-1}] \times \ldots \\ &\times \sum_{i_1=0}^{N_1-1} \exp[2\pi i h_1 i_1/N_1] \rho_s(i_1/N_1, \ldots, i_n/N_n). \end{aligned} \quad (7)$$

Equation (7) forms the basis for the numerical evaluation of the structure factors from a given electron-density function. With a similar discretization of the Patterson function or other functions defined on the unit cell, equation (7) forms the basis for the computation of any Fourier transform that is relevant in nD crystallography.

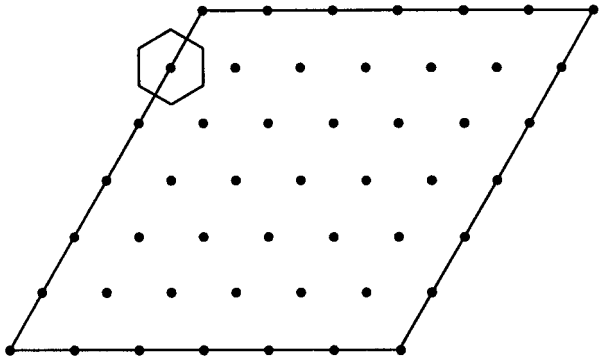

Figure 1
The symmetry-adapted grid on a hexagonal unit cell. The voxels have the shape of a regular hexagon.

3. The *n*-dimensional algorithm

The Beevers–Lipson procedure is based on the observation that the transform of one of the n coordinates is independent of the transform of the other coordinates, as is made explicit by the form of equation (7). For each of the $N_2 \times \ldots \times N_n$ coordinates $(0, i_2, \ldots, i_n)$, the first coordinate of ρ_s is transformed from i_1 space towards h_1 space by a 1D Fourier transform. The resulting function is transformed from i_2 space towards h_2 space for each of the coordinates $(h_1, 0, i_3, \ldots, i_n)$. This process is repeated until the nth coordinate is transformed and $F(\mathbf{H})$ is obtained.

Direct implementation of this procedure results in computer code that is dependent on the number of dimension n. Also, as is suggested by equation (7), the use of an n-dimensional array to store the values ρ_k and $F(\mathbf{H})$ will make the code dependent on n. These problems are overcome if the pixels are uniquely identified by a single integer k that runs from 0 to $N_{\rm pix} - 1$ [equation (1)] and that is used to assign the values of ρ_k to the elements of a 1D array.

Define k by

$$k = i_1 + i_2 N_1 + i_3 N_1 N_2 + \ldots + i_n N_1 \ldots N_{n-1}. \quad (8)$$

For the transformation of coordinate j $(1 \leq j \leq n)$, a total number of

$$P_j = \prod_{\substack{l=1 \\ l\neq j}}^{n} N_l \quad (9)$$

1D Fourier transforms have to be calculated. To define the values of k involved in each of these 1D Fourier transforms, the following sets of numbers are required:

$$S_j = \prod_{l=j+1}^{n} N_l \quad (10)$$

$$s_j = \prod_{l=1}^{j-1} N_l \quad (11)$$

$$\Delta_j = \prod_{l=1}^{j} N_l \quad (12)$$

for $j = 1, \ldots, n$. Note that $P_j = S_j s_j$ for each j. The number of large steps (S_j) with step size Δ_j and the number of small steps (s_j) with step size 1 together define the values of k that indicate the first elements ρ_k involved in the 1D transforms according to

$$k_s = i_S \Delta_j + i_s \quad (13)$$

for $i_S = 0, \ldots, S_j - 1$ and $i_s = 0, \ldots, s_j - 1$. Each 1D transform involves the N_j elements ρ_k with k values defined by

$$k = k_s + i_j s_j \quad (14)$$

for $i_j = 0, \ldots, N_j - 1$.

Equations (8)–(14) form the basis for a modified Beevers–Lipson algorithm:

(i) perform the following operations for each j, starting with $j = 1$ and ending with $j = n$;

(ii) cycle over the large steps;

(iii) cycle over the small steps;

(iv) compute the 1D Fourier transform of the elements defined in equation (14).

It follows that the Beevers–Lipson algorithm for different n has been condensed into a single algorithm that contains n as one of the variables. This has been made possible by explicitly addressing the $N_{\rm pix}$ pixels by a single integer variable [equation (8)].

4. Results and discussion

The Beevers–Lipson procedure leads to a reduction of the number of operations that has to be performed as compared to the straightforward evaluation of the multidimensional Fourier factor. The direct computation of the Fourier transform requires $(N_{\rm pix}N_{\rm ref})$ Fourier factors to be evaluated, where $N_{\rm ref}$ is the number of Fourier coefficients that need to be calculated. For each $j \in \{1, \ldots, n\}$, the Beevers–Lipson procedure requires $N_{\rm pix}/N_j$ 1D Fourier transforms, each of which involves $N_j \log(N_j)$ evaluations of the Fourier factor. The computational complexity of the Beevers–Lipson procedure is thus found to scale as

$$N_{\rm pix} \log(N_{\rm pix}). \quad (15)$$

Compared to the straightforward evaluation of the Fourier transform, the Beevers–Lipson procedure leads to a reduction of the number of operations by a factor of the order of

$$N_{\rm ref}/\log(N_{\rm pix}). \quad (16)$$

It is noticed that equations (15) and (16) are valid exactly, as long as the estimate $N \log N$ for the computational complexity of a 1D FFT is valid. In particular, the Beevers–Lipson procedure propagates the $N \log N$ behaviour of the 1D FFT towards the whole map, independent of the number of dimensions. It follows that the Beevers–Lipson approach leads to a larger reduction of the number of operations with increasing size of the problem.

```
! n                    the dimension.
! ρ(0,···,N_pix − 1)   complex array containing
                       ρ_k.
! f(0,···,N_j − 1)     complex array used for
                       the 1D Fourier transform.
do dimensions j = 1 to n
   do large-step l = 0 to (LS(j) − 1)
   k_s = lΔ(j) − 1
      do small-step s = 0 to (ss(j) − 1)
      k_s = k_s + 1
         do row m = 0 to (N(j) − 1)
         k = k_s + m
         f(m) = ρ(k)
         enddo row
Perform the 1D Fourier-Transform of
f(0,···,N(j) − 1). The result is stored into
f(0,···,N(j) − 1) again.
         do restore m = 0 to (N(j) − 1)
         k = k_s + m
         ρ(k) = f(m)
         enddo restore
      enddo small-step
   enddo large-step
endo dimensions
```

Figure 2
Outline for a computer program for the Beevers–Lipson method in arbitrary dimensions.

Table 1
Computer time in s used for the discrete Fourier transform by the Beevers–Lipson approach.

The values of $N \log(N)$ were scaled by a factor of $K = 420$ ns.

n	$N_{\rm pix}$	$N_1 \times \ldots \times N_n$	Time	$N \log(N)$
3	2^{21}	$128 \times 128 \times 128$	11	
3	2^{24}	$128 \times 128 \times 128$	91	
4	2^{20}	$32 \times 32 \times 32 \times 32$	7	6
4	2^{21}	$64 \times 32 \times 32 \times 32$	12	13
4	2^{22}	$64 \times 64 \times 32 \times 32$	28	27
4	2^{23}	$64 \times 64 \times 64 \times 32$	56	57
4	2^{24}	$64 \times 64 \times 64 \times 64$	118	118
5	2^{24}	$64 \times 32 \times 32 \times 32 \times 32$	125	
6	2^{24}	$16 \times 16 \times 16 \times 16 \times 16 \times 16$	151	

The modified Beevers–Lipson algorithm has been incorporated into a Fortran-90 subroutine, which will be part of a computer program for the maximum-entropy method in nD space. An outline of this routine is given in Fig. 2. The actual computer program was used for a series of test calculations on a SGI Octane computer with ample RAM to retain all arrays in memory, and with a 175 MHz R10000 CPU. The time required for the transform is found to scale with $N_{\rm pix} \log(N_{\rm pix})$ in accordance with (15) (Table 1). The dependence on the dimension n of the time required for the transformation can be explained from the fact that the same value of $N_{\rm pix}$ leads to much smaller N_j when n is increased and that the computational complexity of the 1D transform is greater than $N_j \log(N_j)$ when N_j is small.

In conclusion, we have derived an algoritm based on the Beevers–Lipson procedure that leads to a piece of computer code applicable to Fourier transforms in arbitrary dimensions. The scaling law of the computational complexity [equation (15)] is shown to follow the theoretical prediction. This procedure will find applications in crystallographic studies of quasicrystals and other aperiodic crystals.

Financial support was obtained from the German Science Foundation (DFG) and from the Fonds der chemischen Industrie.

References

Bricogne, G. (1996). *International Tables for Crystallography*, Vol. B, edited by U. Shmueli, pp. 23–106. Dordrecht: Kluwer Academic Publishers.

Dusek, M. & Petricek, V. (1998). *Proceedings of the International Conference on Aperiodic Crystals, APERIODIC '97*, edited by M. de Boissieu, J.-L. Verger-Gaugry & R. Currat, pp. 95–99. Singapore: World Scientific.

Immirzi, A. (1976). *Crystallographic Computing Techniques*, edited by F. R. Ahmed, pp. 399–412. Copenhagen: Munksgaard.
Janssen, T., Janner, A., Looijenga-Vos, A. & de Wolff, P. M. (1992). *International Tables for Crystallography*, Vol. C, edited by A. J. C. Wilson, p. 797. Dordrecht: Kluwer Academic Publishers.
Lorenzo, J. E., Currat, R., Monceau, P., Hennion, B. & Levy, F. (1993). *Phys. Rev. B*, **47**, 10116–10121.
Perez-Mato, J. M. & Elcoro, L. (1994). *J. Phys. I*, **4**, 1341–1352.
Smaalen, S. van (1995). *Crystallogr. Rev.* **4**, 79–202.
Smaalen, S. van & George, T. F. (1987). *Phys. Rev. B*, **35**, 7939–7951.
Walker, M. B. (1986). *Phys. Rev. B*, **34**, 6830–6838