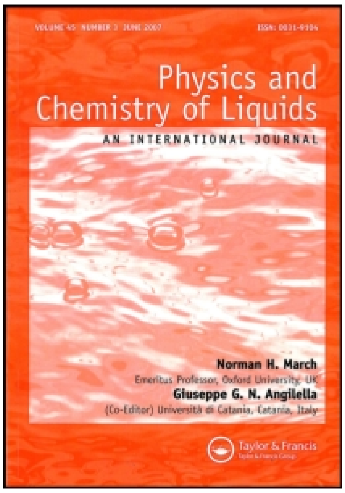
VOLUME 45 NUMBER 3 JUNE 2007
ISSN: 0031-9104
Physics and Chemistry of Liquids
AN INTERNATIONAL JOURNAL
Norman H. March
Emeritus Professor, Oxford University, UK
Giuseppe G. N. Angilella
(Co-Editor) Università di Catania, Catania, Italy
Taylor & Francis

Physics and Chemistry of Liquids
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713646857

A fast fourier transform algorithm for the radial distribution function

A. J. Starshak[a]; R. D. Larsen[a]
[a] Department of Chemistry, Illinois Institute of Technology, Chicago, Illinois

To cite this Article Starshak, A. J. and Larsen, R. D.(1970) 'A fast fourier transform algorithm for the radial distribution function', Physics and Chemistry of Liquids, 2: 1, 45 — 51

To link to this Article: DOI: 10.1080/00319107008084078

URL: http://dx.doi.org/10.1080/00319107008084078

Physics and Chemistry of Liquids. 1970. Vol. 2, pp. 45–51

Printed in Great Britain

A Fast Fourier Transform Algorithm for the Radial Distribution Function

A. J. STARSHAK and R. D. LARSEN

Illinois Institute of Technology
Department of Chemistry
Chicago, Illinois 60616

Received November 15, 1969

Abstract—A simple fast Fourier transformation (FFT) algorithm has been specifically adapted to calculate the experimental radial distribution function. The number of equi-spaced data points must be a power of two [$N = 2^n$ for integer n] and must be greater than the Nyquist frequency [$N = 2(r_{\max})(s_{\max})/2\pi$]. When properly defined, the data set is expanded as an odd function. The greatest advantage of the FFT algorithm is its internal consistency—the ability to exactly transform back to the original domain.

1. Introduction

The history of the development of the "fast Fourier transform" algorithm (FFT) pivots about the early work of Danielson and Lanczos.[1] They showed how to calculate a Fourier transform to obtain a known accuracy from a minimum amount of effort on a desk calculator. Furthermore, they exemplified the theory by calculating the radial distribution function of molten lithium chloride. Now that their algorithm has been adapted to digital computers and popularized by Cooley and Tukey, Sande, and others,[2,3] its use has revolutionized many other areas of Fourier analysis. Thus, a reexamination of its utility in calculating the experimental radial distribution function of liquids is timely.

The main topic of this paper is the calculation of the radial distribution function by means of a small FFT.[4] However, this paper will also briefly contrast two other methods of calculating the central Fourier transform of liquid diffraction analysis. The first is a trapezoidal approximation to the diffraction integral, similar to that used by many investigators. The second is a discrete summation routine adapted from Guinier[5] and representative of a number of summation techniques. Neither of these two other methods has been optimized; they are simply included for comparison purposes. The average times for the transformations and their inverses are given in Table 1. But far more important, the internal consistency—the ability to accurately transform back to the original domain again—is given in Table 2 and Fig. 1.

TABLE 1

	Run Times (Sec)			
$N =$	64	128	192	256
I (Trapezoidal)	24.0	102.7	213.6	380.0
II Summation)	6.2	25.0	56.5	101.5
III (FFT)	1.0	4.3	18.8	38.5

All times are relative and the average of the Fourier transformation and its inverse on an IBM 360/40. The times include all peripheral calculations to go from {si(s)} to $\{rD(r)\}$ or from $\{rD(r)\}$ to $\{\widehat{si}(s)\}$.

In the theory of liquid diffraction analysis, the Fourier integral can be written:

$$rD(r) = \frac{2}{\pi}\int_0^\infty si(s)\sin rs\,ds \tag{1}$$

where $i(s)$ is the intensity of X-radiation scattered at $s = 4\pi/\lambda^{-1}\sin\theta$, properly corrected for polarization, incoherent scattering, etc., and normalized to electron units.[6]

The evaluation of such an integral on a digital computer involves at least two approximations. The first is the approximation of the infinite integral by a finite data set:

$$rD(r) = \frac{2}{\pi}\int_0^{s\,\max} si(s)\sin rs\,ds \tag{2}$$

The magnitude of this approximation is proportional to the value of $si(s_{\max})$. The second is the numerical method used to approximate a continuous integral by a discrete quadrature:

$$r_nD(r_n) = \frac{2\Delta s}{\pi}\sum_{k-1}^{N-1} s_k i(s_k)\sin r_n s_k \tag{3}$$

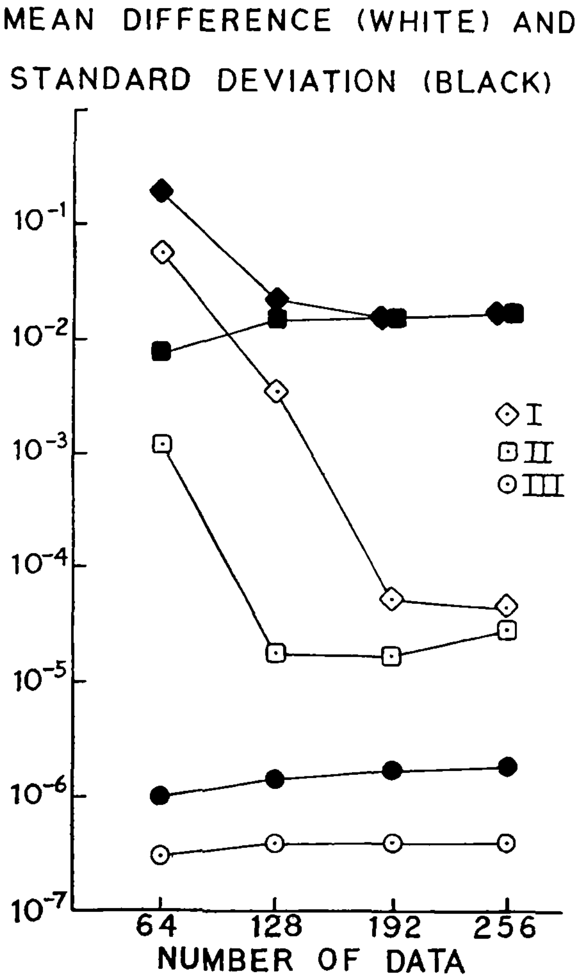

Figure 1

It is in this context of numerical approximations that the FFT has such remarkable properties, for the algorithm provides a prescription to expand a discretely sampled continuous function over a complete set of orthonormal exponential functions. The Fourier coefficients so obtained reconstitute the discretely sampled function exactly and provide for a least-square interpolation formula for all intervening points. To illustrate, therefore, both the speed and accuracy of the FFT, we first consider for comparison two other algorithms.

2. Other Transformation Algorithms

The first method of numerical approximation is a trapezoidal integration algorithm (I) adapted for non-equispaced increments:

$$\text{(I)}\qquad r_nD(r_n) = \frac{2}{\pi}\sum_{k=1}^{N} s_k i(s_k)\sin r_n s_k\,\Delta s_k,\quad n = 1, 2 \ldots N \qquad (4)$$

where $\Delta s_k = (s_{k+1} - s_{k-1})$ for $2 \leqslant k \leqslant N-1$.

A second approximation method (II) adapted from Guinier[5] has an advantage in being specifically designed for discrete summations but has the disadvantage of requiring equispaced data points. (The non-equispaced data are interpolated by means of a Lagrangian interpolation formula.) This approximation can be written as:

$$\text{(II)}\qquad r_nD(r_n) = \frac{2\Delta s}{\pi}\sum_{k=1}^{N} s_k i(s_k)\sin r_n s_k,\quad n = 1, 2, \ldots N \qquad (5)$$

where $\Delta s = (s_N - s_1)/(N-1)$.

3. Fast Fourier Transform Algorithm

The appropriate formula for the FFT in the liquid diffraction cortext is:

$$\text{(III)}\qquad r_nD(r_n) = \frac{2\Delta s}{\pi}\sum_{k=0}^{2N-1} s_k' i(s_k')\, e^{+2\pi ink/2N},\quad n = 0, 1 \ldots N-1 \qquad (6)$$

where $\Delta s = (s_N - s_1)/N$. Observe that the complex exponential transform is used here in contrast to the Danielson–Lanczos presentation which used just the sine transform. The use of all N roots of unity around the complex unit circle is an essential element in both the theory and execution of the FFT.[3]

In implementing the FFT, however, additional problems arise. Not the least of these is the fact that linear frequency (not radial frequency) is involved: $s' = 2\lambda^{-1}\sin\theta = s/2\pi$. Secondly, there are strict limitations on the number of equispaced data points required to give the desired transform. Finally, the waveform must be expanded as an odd function. How each one of these factors is treated is briefly described below.

The first difficulty with the FFT is the use of $s' = 2\lambda^{-1}\sin\theta = s/2\pi$. Although it is a simple matter to divide $s_k i(s_k)$ by 2π, the argument of the exponential is changed, making it very sensitive to the density of data points used to describe the intensity function.

The second problem concerns the number of data points. The simple form of the FFT[4] which was used requires a number of points equal to a power of two: $N = 2^n$. This number is intimately connected with both the length of the original data set and the desired length of the transform by means of the Nyquist frequency. In the data set used,[7] the maximum value of s was $s_{\max} = 15.824 \approx 15.708 = 5\pi$. The Nyquist relation states that the sampling must be at least twice the highest frequency:

$$N = 2(r_{\max})(s_{\max})/2\pi \tag{7}$$

So, with $N = 2^6 = 64$, the maximum value of r was $r_{\max} = 12.8$ A, giving $\Delta r = 0.2$ A.

By simply increasing the record length to $N = 2^7 = 128$ and keeping $s_{\max} = 15.708\,\mathrm{A}^{-1}$, the same Δr is obtained, except $r_{\max}$ is increased to 25.6 A, extended with (noisy) zeros. Accordingly, by adding on zeros to double $s_{\max}$ to $31.416\,\mathrm{A}^{-1}$, the number of data points defining $rD(r)$ can be doubled. (The process can be continued for $N = 2^8 = 256$, but 768 zeros must be attached to the $si(s)$ curve to increase the density of data points for $rD(r)$. In the end, $N = 256$ requires a transform of 2048 data points.)

When the input data set is properly defined, the function can be expanded as an odd function, not from $-N \leqslant k \leqslant N-1$, but from $0 \leqslant k \leqslant 2N-1$. Thus, the transformation is actually that of the negative of a normal expansion, requiring a negative compensation at the output.

4. Discussion of Transformation Algorithms

The data set used for these comparisons was taken from a report of M. D. Danford.[7] The original 121 values of $I(s)$ were measured from $s = 1.233$ ($\theta = 4.0°$) to $s = 15.824$ ($\theta = 63.5°$). These values were: (1) extrapolated so that $0.0 \leqslant s \leqslant 16.0$, (2) normalized to electron units,[6] (3) then interpolated to give equispaced data points at densities of 64, 128, 192, and 256 per $15.708\,\mathrm{A}^{-1}$. All three methods used the same input data, so that the differences are characteristic of the algorithms alone.

The results of the time-trials are shown in Table 1. These times, which include all peripheral calculations necessary to transform from equispaced $\{s, si(s)\}$ to $\{r, rD(r)\}$ and back, are the averages of the transformation and its inverse. The adjective "fast" is appropriate for the FFT, especially when considering that an extended number of 512 and 2048 data points had to be transformed to get a radial distribution function for $N = 128$ and 256, respectively.

The relative accuracies of the algorithms are shown in Table 2. The set of N equispaced $\{s, si(s)\}$ was transformed to $\{r, rD(r)\}$ and back to $\{s, \widehat{si}(s)\}$. The mean difference $N^{-1}\sum[si(s)_j - \widehat{si}(s)_j]$ and the standard deviation $(N-1)^{-1}\sum[si(s)_j - \widehat{si}(s)_j]^2$ are reported. The limit of accuracy of the IBM 360/40 used is about 6.5 significant figures for simple additions, so deviations greater than 1×10^{-6} can be attributed to the algorithm, as seen in Fig. 1.

TABLE 2

Accuracy of Transform

$N =$	64	128	192	256
I	4.7×10^{-2}	3.4×10^{-3}	5.0×10^{-5}	4.2×10^{-5}
(Trapezoidal)	$\pm 1.9 \times 10^{-1}$	$\pm 2.2 \times 10^{-2}$	$\pm 1.5 \times 10^{-2}$	$\pm 1.6 \times 10^{-2}$
II	1.2×10^{-3}	1.8×10^{-5}	1.7×10^{-5}	2.8×10^{-5}
Summation)	$\pm 7.9 \times 10^{-3}$	$\pm 1.5 \times 10^{-2}$	$\pm 1.5 \times 10^{-2}$	$\pm 1.6 \times 10^{-2}$
III	0.3×10^{-6}	0.4×10^{-6}	0.4×10^{-6}	0.4×10^{-6}
(FFT)	$\pm 1.0 \times 10^{-6}$	$\pm 1.4 \times 10^{-6}$	$\pm 1.7 \times 10^{-6}$	$\pm 1.8 \times 10^{-6}$

The mean difference and standard deviation between the original $\{si(s)\}$ and twice transformed $\{\widehat{si}(s)\}$ are presented. Values greater than 1×10^{-6} can be attributed to the algorithm.

The mean difference for a simple trapezoidal integration (I) becomes acceptable at $N = 192$ data points; the Guinier summation algorithm (II) has an acceptably small mean difference when $N = 128$ data points. However, the confidence level for both would be rather low with a standard deviation $\mathcal{O}(10^{-2})$. The mean difference of the FFT employed (III) is uniformly $\mathcal{O}(10^{-7})$ with a standard deviation $\mathcal{O}(10^{-6})$. This great reliability is perhaps the most important feature of the algorithm. As the complex exponential $e^{+2\pi i n k/2N}$ is a complete set (modulo $2N$) for $0 \leqslant n, k \leqslant 2N-1$, we observe the fact that a discretely sampled function can be expanded exactly over a complete orthonormal basis set.

This order of internal consistency allows the transformation from one domain to the other almost indiscriminately. Thus, a whole class of deconvolution procedures is possible with this fast Fourier transformation algorithm.

REFERENCES

1. Danielson, G. C. and Lanczos, C., *J. Franklin Inst.* **233,** 365, 435 (1942).
2. Cooley, J. W. and Tukey, J. W., *Math of Comput.* **19**, 297 (1965); Gentlemen, W. M. and Sande, G. in 1966 Fall Joint Computer Conf., *AFIPS Proc.* **29**, 563 (1966).
3. Cochran, W. T., *et al.*, *IEEE Trans* **AU-15**, 45 (1967).
4. Robinson, E. A., *Multichannel Time Series Analysis with Digital Computer Programs* (Holden–Day, Inc., San Francisco, 1967), p. 63.
5. Guinier, A., *X-ray Diffraction* (Freeman and Co., San Francisco, 1963), p. 364.
6. Pings, C. J. and Waser, J., *J. Chem. Phys.*, **48**, 3016 (1968).
7. Danford, M. D., *ORNL-4244*, 15 (1968).