METEOROLOGICAL OFFICE

Boundary Layer Branch (Met O 14)

Turbulence and Diffusion Note No. 9.

SPECTRAL ANALYSIS USING THE FAST FOURIER TRANSFORM (FFT)

By D. Rayment.

Note As this paper has not been published, permission to quote from it should be obtained from the Head of the above Branch of the Meteorological Office.

MCB

CONTENTS

Turbulence and Diffusion Note No.9

SPECTRAL ANALYSIS USING THE FAST FOURIER TRANSFORM(FFT)

by D. Rayment

Introduction

1. The FFT is a numerical technique for rapidly computing Fourier coefficients. Suppose we have $N = 2^m$ data points. About N^2 arithmetic operations are required to find the Fourier coefficients using direct methods (i.e. each of the N data points has to be multiplied by a trigonometric function to obtain each of the N Fourier coefficients). With the FFT the number of operations is reduced to about $2N\log_2 N$ which increases the speed of computation, for say 1024 data points, by about 50 times. An actual evaluation, using the FFT, for 8192 points took 5 seconds compared with about ½ hour by conventional techniques (Cochran 1967).

When N is composed of factors 2, 3 or 4 the maximum computational advantage is realised and such factoring can nearly always be arranged by adding zeros to the data. Because the number of operations involved, in using the FFT, is smaller,round off errors will be much reduced.

This great increase in speed makes it preferable to compute spectra directly from the Fourier coefficients rather than by the "lagged products" method or some digital filtering method. In addition if one requires the variance or covariance of a function(s) it is usually faster to calculate these from the 'power spectrum' or 'cospectrum' (Tukey 1966).

Derivation of Spectral Formulae in terms of Fourier coefficients

2. Any periodic function, $x(t) = x(t+T)$, where T is the period, can, with some restrictions, be represented by a Fourier series:-

$$x(t) = a_o/2 + \sum_{k=1}^{\infty} (a_k \cos 2\pi kt/T + b_k \sin 2\pi kt/T) \qquad (1)$$

In complex notation (1) can be written:-

$$x(t) = \sum_{k=-\infty}^{\infty} C_k \exp(-2\pi ikt/T) \qquad (2)$$

so that

$$C_k = \frac{1}{T}\int_o^T x(t) \exp(2\pi ikt/T)\, dt \qquad (3)$$

and

$$C_k = (a_k + ib_k)/2 \ , \ C_{-k} = (a_k - ib_k)/2 \text{ and } C_o = a_o/2. \qquad (4)$$

3. Defining a second function

$$y(t) = \sum_{k=-\infty}^{\infty} D_k \exp(-2\pi ikt/T)$$

where $D_k = (c_k + id_k)/2$ etc., the average cross-correlation, $\sigma_{xy}(\tau)$, where $x(t)$ and $y(t)$ shall now have zero mean (i.e. $a_o = c_o = 0$), is given by

$$\sigma_{xy}(\tau) = \frac{1}{T}\int_o^T x(t)\, y(t-\tau)\, dt \tag{5}$$

where τ is the lag of one function on the other along the t axis and $\sigma_{xy}(\tau)$ is periodic in T.

Substituting for $y(t - \tau)$ in (5) gives

$$\sigma_{xy}(\tau) = \frac{1}{T}\int_o^T x(t) \sum_{k=-\infty}^{\infty} D_k \exp(-2\pi ik(t-\tau)/T)\, dt \tag{6}$$

Treating k and t as independent (6) can be written as

$$\sigma_{xy}(\tau) = \sum_{k=-\infty}^{\infty} D_k \exp(2\pi ik\tau/T)\, \frac{1}{T}\int_o^T x(t) \exp(-2\pi ikt/T)\, dt$$

$$= \sum_{k=-\infty}^{\infty} D_k C_{-k} \exp(2\pi ik\tau/T) \tag{7}$$

Since the Fourier coefficients a_k and b_k etc., as in (1) are not defined for -k we write (7) as

$$\sum_{k=o}^{\infty} \left[D_k C_{-k} \exp(2\pi ik\tau/T) + D_{-k}C_k \exp(-2\pi ik\tau/T)\right] \tag{8}$$

Using the fact that

$$D_kC_{-k} + D_{-k}C_k = (a_kc_k + b_kd_k)/2$$

and

$$D_kC_{-k} - D_{-k}C_k = -i(b_kc_k - a_kd_k)/2$$

we derive finally

$$\sigma_{xy}(\tau) = \frac{1}{2}\sum_{k=o}^{\infty} \left[(a_kc_k + b_kd_k)\cos(2\pi k\tau/T) + (b_kc_k - a_kd_k)\sin(2\pi k\tau/T)\right] \tag{9}$$

4. The cross-correlation at zero lag is:-

$$\sigma_{xy}(0) = \overline{x(t)y(t)} = \frac{1}{2}\sum_{k=o}^{\infty}(a_k c_k + b_k d_k) \qquad (10)$$

which in integral form can be written

$$\overline{x(t)y(t)} = \int_o^{\infty} S(n)_{xy}\,dn \qquad (11)$$

where n is the frequency.

5. The power spectrum gives the contribution to the variance at different frequencies. So we have

$$\sigma_x^{\ 2} = \frac{1}{T}\int_o^T \left[x(t)\right]^2 dt = \frac{1}{2}\sum_{k=o}^{\infty}(a_k^{\ 2} + b_k^{\ 2}) \qquad (12)$$

and $$\sigma_x^{\ 2} = \int_o^{\infty} S(n)_x\,dn = \int_o^{\infty} nS(n)_x\,d(\log n) \qquad (13)$$

6. In general we can write

$$\sigma_{xy}(\tau) = \int_{-\infty}^{\infty}\phi_{xy}(n)\exp(2\pi i n\tau)\,dn \qquad (14)$$

where n = the frequency and $\phi_{xy}(n)$ is the complex cross-spectrum.

We write $\phi_{xy}(n) = Co(n) + iQ(n)$

where Co(n) is called the cospectrum and measures the contribution to the covariance in each frequency band and Q(n) is the quadrature-spectrum which measures the $\pi/2$ out of phase relationship between the two series at each frequency (Jones 1958).

Now Co(n) is an even function of n and Q(n) is an odd function of n - this follows from the definitions that

$$Co(n) = \int_o^{\infty}\tfrac{1}{2}(\sigma(\tau) + \sigma(-\tau))\cos 2\pi n\tau\,d\tau$$

and $$Q(n) = \int_o^{\infty}\tfrac{1}{2}(\sigma(\tau) - \sigma(-\tau))\sin 2\pi n\tau\,d\tau$$

We have, therefore, expanding (14) and changing the limits

$$G_{xy}(\tau) = \int_{o}^{\infty}(2Co(n)\ \cos 2\pi n\tau - 2Q(n)\ \sin 2\pi n\tau)dn$$

Letting $n = k/T = k\Delta n$ the summation version of this becomes:-

$$G_{xy}(\tau) = \frac{1}{T}\sum_{k=o}^{\infty}\left[S(k\Delta n)\ \cos 2\pi k\tau/T \ - q(k\Delta n)\ \sin 2\pi k\tau/T\right]$$

where $2Co(n) \equiv S(k\Delta n)$ and $2Q(n) \equiv q(k\Delta n)$, i.e. if only positive frequencies are considered the cospectrum is $2Co(n)$ and the quadrature spectrum is $2Q(n)$. Comparing with (9) gives

$$\left.\begin{aligned} S(k\Delta n) &= \tfrac{T}{2}(a_k c_k + b_k d_k) && \text{i.e. the cospectrum points and} \\ q(k\Delta n) &= \tfrac{T}{2}(a_k d_k - b_k c_k) && \text{i.e. the quadrature spectrum points} \end{aligned}\right\} (15)$$

Summation versions of the Spectral Formulae

7. For a discrete, equally spaced time series, of zero mean, where t = time, N = the number of data points and n = a frequency, writing

$t = j\Delta t$, $(j = 0,1,- - - N-1)$, $T = N\Delta t$ and $n = k/T = k\Delta n$, $(k = 0,1,- - N-1)$, ignoring end corrections the summation version of (3) is:-

$$C(k\Delta n) = \frac{1}{N}\sum_{j=o}^{N-1} x(j\Delta t)\ W^{jk} \qquad (16)$$

where $W = \exp(2\pi i/N)$ and $C(k\Delta n)$ is used to replace C_k to indicate that the Fourier coefficients are evaluated at frequency intervals of Δn starting at zero frequency.

Equation (16) has an exact Fourier transform:-

$$x(j\Delta t) = \sum_{k=o}^{N-1} C(k\Delta n)\ W^{-jk} \qquad (17)$$

8. It can be seen now that the physical quantity we are considering is replaced by a limited series of data 'spikes' separated by time intervals of Δt. This has two effects on the spectra called "leakage" and "aliassing".

9. Leakage is caused by the presence, in the physical quantity of frequencies other than those at $k\Delta n$. If the quantity really consisted of a series of spikes (16) would give the precise Fourier coefficients. However these inbetween frequencies modify the coefficients over a large range of adjacent frequencies. This range can be reduced, at the expense of statistical accuracy, by using 'data windows' or by applying, for example, 'hanning' to the derived coefficients. For a full discussion see Blackman and Tukey 1958 and Bingham 1967.

10. Aliassing is caused by there being significant contributions to the variance or covariance beyond the upper frequency limit of a particular analysis. This upper limit is determined by the sampling rate.

In equation (16), because $W^{jk} = W^{-j(N-k)}$, useful coefficients are only produced up to a frequency of $N\Delta n/2$ the rest being merely a set of reflected values. It should be noted that equation (16) only looks at positive frequencies and the results C(1) to C(N-1) are symmetric about C(N/2). So that if there are contributions, in the data, beyond $N\Delta n/2$, these will be added to the coefficients corresponding to frequencies below $N\Delta n/2$, i.e. aliassing will occur.

If the sampling rate cannot be increased sufficiently the total spectrum cannot be resolved. However aliassing can be removed by using a physical or mathematical low pass filter.

11. Other losses can be caused by inadequate instrumental response — the effect being that of a low pass filter, and/or too short a recording period — the effect being that of a high pass filter.

12. The FFT provides a rapid evaluation of the summation part of (16), i.e.

$$A(k\Delta n)_x = \sum_{j=o}^{N-1} x(j\Delta t)\, W^{jk} \qquad (18)$$

The following notation will be used

$$A(k\Delta n)_x = a_{Fx}(k\Delta n) + ib_{Fx}(k\Delta n)$$

we have $A(k\Delta n)_x = NC(k\Delta n) = \frac{N}{2}(a'_k + b'_k)$ so that

$$a_{Fx}(k\Delta n) = a'_k \, N/2 \text{ etc.,} \tag{19}$$

where a'_k and b'_k are the aliassed and leaked versions of a_k and b_k.

Various notations will now be taken to be equivalent for example

$x(j\Delta t) = x(j) = x_j$, $a_F = a_F(k\Delta n)$, etc.

13. The coefficients a_F and b_F etc., can now be substituted into equations (12), (13) and (15) where $k = 0,1, - - N/2$, this gives the aliassed '<u>cospectrum</u>' points

$$S(k\Delta n)_{xy} = \frac{2\Delta t}{N}(a_F c_F + b_F d_F) \tag{20}$$

and '<u>quadrature spectrum</u>' points

$$q(k\Delta n)_{xy} = \frac{2\Delta t}{N}(a_F d_F - b_F c_F) \tag{21}$$

and '<u>power spectrum</u>' points

$$S(k\Delta n)_x = \frac{2\Delta t}{N}(a_F{}^2 + b_F 2) \tag{22}$$

14. These formulae can be derived directly by substituting (16) and (17) into (5) so that instead of (6) there is

$$\sigma_{xy}(L\Delta t) = \frac{1}{N}\sum_{j=o}^{N-1}\sum_{k=o}^{N-1} C_{(N-k)}\, W^{jk} \sum_{k=o}^{N-1} D_k \, W^{-jk}\, W^{Lk} \tag{23}$$

where $\tau = L\Delta t$ and

$$\sum_{k=o}^{N-1} C_k \, W^{-jk} = \sum_{k=o}^{N-1} C_{(N-k)}\, W^{jk}$$ has been used.

The orthoganality relations

$$\sum_{j=o}^{N-1} W^{j(r-s)} \begin{array}{l} = N \text{ if } r = s \\ = 0 \text{ if } r \neq s \end{array}$$

where $|r - s| \leq m$, permits us to write

$\sigma_{xy}(L\Delta t) = \sum_{k=o}^{N-1} C_{(N-k)} D_k \, W^{Lk}$ which gives all the discrete spectral formulae.

In passing, from the last quoted formula, it is clear that if one requires cross-correlations and/or correlation coefficients and/or autocorrelations one can perform a FFT treating $C_{(N-k)}D_k$ as the data input in place of $x(j\Delta t)$ and W^{Lk} as the complex multiplier in place of W^{jk} in (18). This effectively gives circular cross- and autocorrelations and as many zeros need to be added to the $C_{(N-k)}D_k$ data as the number of L's that are used. A full discussion is given by Gentleman and Sande 1966.

15. The $S(k\Delta n)$ give values at frequency intervals $\Delta n = 1/T$ where $S(0) = 0$ and $S(N\Delta n/2)$ is the estimate at the frequency $N\Delta n/2$, called the Nyquist frequency, f_N. The Nyquist frequency is related to the sampling rate by $f_N = 1/(2\Delta t)$.

Statistical Reliability of Estimates

16. Apart from round off errors, the square root of the mean square fractional error ϵ, for a Gaussian process, in $S(k\Delta n)_x$ is given by

$$\epsilon \simeq \sqrt{1/(T\Delta n)} \qquad \text{[Richards 1967]} \qquad (24)$$

so that individual power spectrum estimates given by (22) have a fractional error of 1 (i.e. 100% error).

To make the estimates more reliable we can take an average of adjacent $S(k\Delta n)$'s or average over sets of $S(k\Delta n)$'s for different (maybe overlapping) record lengths. For example if we average over p adjacent $S(k\Delta n)$'s then Δn will become p/T and the fractional error is $\sqrt{1/p}$. Both types of averaging may be used.

Equation (24) has only been used as a guide here since the time series will probably not have come from a Gaussian process. The errors in the other types of spectra are more complicated though similar averaging techniques can be employed.

If we assume $nS(n) \propto n^r$ then mean estimates are best made from the mean log of $nS(n)$'s which can then be plotted at the mean log frequency of the range used. In practice this method of averaging is rather cumbersome and one usually has a sufficient number of points to use 'short section' linear averaging, which only distorts the shape slightly, and yet still preserves adequate statistical stability. This is particularly true of meteorological spectra which tend to be fairly gently curving. Smoothing through very sharp peaks can be quite troublesome and a method, using a triangular weighting function, is described by Singleton and Poulter 1967.

Referring to (13), areas will be preserved if nS(n) is plotted against log n.

The Fast Fourier Transform

17. Consider $N = 2^m$ data points and (18) written as

$$A(k)_x = \sum_{j=o}^{N-1} x(j)\, W^{jk} \qquad (25)$$

Following Cooley and Tukey 1965, j and k can be recast as binary numbers:-

$$\left.\begin{aligned} j &= j_{m-1}2^{m-1} + j_{m-2}2^{m-2} + \ldots\ldots + j_o \\ \text{and} \qquad k &= k_{m-1}2^{m-1} + k_{m-2}2^{m-2} + \ldots\ldots + k_o \end{aligned}\right\} \qquad (26)$$

So that x(j) can be taken to represent the data points successively in time (these indexes can also represent successive storage locations in a computer) and A(k) will characterise the Fourier coefficients successively in the frequency domain. To illustrate the indexing we have for example

$A(0,0,0) = A(0)$, $A(0,0,1) = A(1) = a_F(1) + ib_F(1)$, $x(0,1,1) = x_3$ etc.

By inserting (26) into (25) we derive

$$A(k_{m-1}, k_{m-2}, \ldots\ldots k_1, k_o) = \sum_{j_o=o}^{1} \ldots\ldots \sum_{j_{m-1}=o}^{1} x(j_{m-1}, \ldots\ldots, j_o)\, W^{k_o j_{m-1} 2^{m-1}} \times \ldots\ldots \times W^{(k_{m-1}2^{m-1} \ldots + k_o) j_o} \qquad (27)$$

This process has two important consequences. First it has removed a large number of W terms. For example

$W^{k j_{m-1} 2^{m-1}}$ has been replaced by $W^{k_o j_{m-1} 2^{m-1}}$ since these are equal.

In general

$$W^{kj_{m-s}2^{m-s}} = W^{(k_{s-1}2^{s-1} +.....+k_o)j_{m-s}2^{m-s}} \qquad (28)$$

To prove this expand k as follows

$$k = (k_{m-1}2^{m-1}+k_{m-2}2^{m-2}+...+k_{m-r}2^{m-r}+..+k_s2^s+k_{s-1}2^{s-1}+...+k_o)$$

Consider now the terms that are removed from the left hand side of (28) i.e.

$$W^{(k_{m-1}2^{m-1}+..+k_{m-r}2^{m-r}+...+k_s2^s)j_{m-s}2^{m-s}} \qquad (29)$$

Remembering that $W = \exp(2\pi i/N) = \exp(2\pi i/2^m)$ (29) becomes

$$\exp(2\pi i(k_{m-1}2^{m-1-s}+k_{m-2}2^{m-2-s}+...+k_{m-r}2^{m-r-s}+...+k_s2^o)j_{m-s}) \qquad (30)$$

now $s \leqslant m-r$ therefore (30) equals unity and (28) is proven.

18. The second important factor can best be illustrated by expanding (27) for the case where N = 8 (i.e. m = 3). (27) then becomes

$$A(k_2,k_1,k_o) = \left[\left\{(x_o+x_4\, W^{4k_o}) + \left[(x_2+x_6\, W^{4k_o})\, W^{2k_o}\right] W^{4k_1}\right\}\right]$$
$$+ \left[\left\{(x_1+x_5\, W^{4k_o})\, W^{k_o} + \left[(x_3+x_7\, W^{4k_o})\, W^{3k_o}\right]\, W^{4k_1}\right\} W^{2k_1}\right] W^{4k_2} \qquad (31)$$

Notice that a change in the j_{m-1} from 0 to 1 produces a jump in the x data subscript of N/2 and also note that

$$W^{4k_o} = 1 \text{ for } k_o = 0 \text{ and } -1 \text{ for } k_o = 1.$$

Starting with data in normal order, $x_o,x_1,...x_7$, the inner brackets are multiplied by a W with a k_o index and produce a set of 8 (i.e. N) partial results

$X_o,X_1,...X_7$, where $X_o = x_o+x_4$, $X_1 = x_o-x_4$,.....$X_6 = x_3+x_7$, and $X_7 = (x_3-x_7)W^3$.

The next set of terms involving k_1 indexes can now be worked out by bringing together the <u>same</u> set (in terms of subscripts) of X's but multiplied by a different W index. It is this property which makes computation easy. To illustrate this clearly let

$$w = \exp(\pi i) = W^{2^{m-1}}$$

so that the set of X's in terms of x's become

$$x_o+x_4,(x_o-x_4)w^o,x_1+x_5,(x_1-x_5)w^{1/2^{m-1}},x_2+x_6,(x_2-x_6)w^{2/2^{m-1}},x_3+x_7,(x_3-x_7)w^{3/2^{m-1}}$$

i.e.

$$X_o\ ,\ X_1\quad ,\ X_2\ ,\ X_3\quad ,\quad X_4\ ,\ X_5\quad ,\quad X_6\ ,\ X_7$$

This process will have taken the data in one storage area and read the partial results into another area. The w index is in fact

$$(j/2^{p-1})_I(1/2^{m-p})$$

and provides a complex multiplication of every $(x_j - x_{j+N/2})$ where j enumerates the x subscript, I means take integer part of first bracket and p is the pass number. So that in the example above, which was the first transfer p = 1. The next transfer which reads results back into the first storage area, gives

$$X_o+X_4,(X_o-X_4)w^o,X_1+X_5,(X_1-X_5)w^o,X_2+X_6,(X_2-X_6)w^{1/2^{m-2}},X_3+X_7,(X_3-X_7)w^{1/2^{m-2}}$$

i.e. p = 2. The whole computational procedure can be summed up with two formulae

$$X_{2j} = x_j + x_{j+N/2} \text{ and } X_{2j+1} = (x_j - x_{j+N/2})\exp\left[\pi i(j/2^{p-1})_I(1/2^{m-p})\right] \qquad (32)$$

Note that the j subscript only runs from 0 to $\frac{N}{2} - 1$ because the partial results are taken in pairs.

This method is given by Singleton 1967.

19. In the above example one further pass produces the Fourier coefficients and in general m passes have to be made. The coefficients arrive in a scrambled form, called 'binary inverted order', and the FFT computer programme, given in the appendix, includes a 'reshuffle routine' to produce the coefficients in serial order i.e.

$a_o, a_1, \ldots\ldots a_{N-1}$ and $b_o, b_1, \ldots\ldots b_{N-1}$.

Formulae (32) transfers back and forth between storage areas in the computer and for 'raw' use of the FFT four storage areas of length N must be used. For example A stores for the data, B stores for a set of zeros (imaginary data), C and D stores for transferring. To arrive at the scrambled coefficients as we have said, m passes have to be made, and a further N/4 passes have to be done to unscramble these (see flow graph in the appendix).

Improvements on the 'raw' use of the FFT

20. If we have two sets of data x(j) and y(j) of length N we can compute the useful Fourier coefficients together by replacing x(j) in (25) by x(j) +iy(j) giving

$$A(k)_{xy} = \sum_{j=o}^{N-1} (x(j) + iy(j))\, W^{jk} \qquad (33)$$

By putting $g = 2\pi/N$ (33) becomes

$$A(k)_{xy} = \sum_{j=o}^{N-1} \left[(x_j \cos gjk - y_j \sin gjk) + i(y_j \cos gjk + x_j \sin gjk)\right] \qquad (34)$$

To recover $A(k)_x$ and $A(k)_y$ we require a summation of the form

$$\sum_{j=o}^{N-1} \left[(x_j \cos gjk + ix_j \sin gjk) + i(y_j \cos gjk + iy_j \sin gjk)\right] \qquad (35)$$

However the FFT answers will be in the form corresponding to (34). Making use of the fact that

$$\cos gj(N-k) = \cos gjk$$

and $\quad \sin gj(N-k) = -\sin gjk$

we can easily show that

$$A(k)_x = \mathrm{Re}\left[A(k)_{xy} + A(N-k)_{xy}\right]/2 + i\,\mathrm{Im}\left[A(k)_{xy} - A(N-k)_{xy}\right]/2 \quad \text{and}$$

$$A(k)_y = \mathrm{Im}\left[A(k)_{xy} + A(N-k)_{xy}\right]/2 - i\,\mathrm{Re}\left[A(k)_{xy} - A(N-k)_{xy}\right]/2 \qquad (36)$$

This process, the so called 'dual use', produces the coefficients

$A(1)_x$ and $A(1)_y$ up to $A(\frac{N}{2} - 1)_x$ and $A(\frac{N}{2} - 1)_y$.

21. If N is even (25) can be written as

$$A(k)_x = \sum_{v=0}^{\frac{N}{2}-1} x(2v)\, W^{2vk} + W^k \sum_{v=0}^{\frac{N}{2}-1} x(2v+1)\, W^{2vk} \qquad (37)$$

This provides two transforms, one for even and one for odd numbered data. By treating the even data as real and the odd data as imaginary we can perform the 'dual use' method and recombine by multiplying the odd data coefficients by W^k. This method (Bingham 1967) uses only half the storage space and takes approximately half the time compared with the raw use of the FFT.

APPENDIX I

Spectral Results using the FFT

Starting with a set of coded, digitized data the sort of process required, to derive spectra, is illustrated by the flow diagram in Fig.A, page 5. Step 2 could be replaced by several other techniques e.g. fitting a low frequency sine wave or polynomial or even a straightforward mean. Also if the data is extracted from a filter unit, which has a zero mean output, step 2 could be omitted altogether. No 'cast iron' rules can be laid down since it depends not only on the original data but also on the type of spectra one is dealing with.

"Hanning", if it is to be applied, can be inserted either between steps 2 and 3 or sometimes between steps 3 and 4. Thus if one has spectra that are multi-peaked and only the relative values of peaks are of interest the following formulae can be applied;-

replace x_j by $x'_j = \frac{1}{2}(1 - \cos(2\pi j/N))x_j$ (38)

Putting this expression into (18) and expanding gives a set of "hanned" Fourier coefficients:-

$$A'_k = \tfrac{1}{4}(-A_{k-1} + 2A_k - A_{k+1}) \qquad (39)$$

Whether one applies equation (38) between steps 2 and 3 or (39) between steps 3 and 4 is just a computational detail. These two 'hanning' formulae do not however preserve the true amplitude of the spectral points and should not be applied where the area under the spectral curve is important. Meteorological spectra, which are often broad single peaks, have areas which directly relate to fluxes or variances of quantities and the aforementioned 'hanning' formulae should not be used. Some 'hanning' could be tried however by, for example, applying in place of (38):-

$$x'_j = \tfrac{1}{2}(1 - \cos(8\pi j/N))x_j \qquad (40)$$

where this is only used to multiply the first and last eighth of data the rest being multiplied by unity, so that spectral amplitudes are only slightly altered.

No simple and usable equivalent to (40) in the frequency domain (as (39) to (38)) exists. From meteorological spectra that have been worked out, it seems unnecessary to apply 'hanning'.

At step 4 the Fourier coefficients are simply substituted in the appropriate formulae given earlier and in fact only the first (N/2) + 1 results will be useful. The averaging used at step 5 has already been discussed.

Comparison of FFT Spectra and Spectra derived using a set of Averaging times

Experiments have been performed at Cardington on a 28 ft mast using the Cardington turbulence probe and data tapes of ϕ, the inclination of the wind to the horizontal, T, the temperature and V, the total wind fluctuations, have been produced. These were processed on the Mercury computer at the Chemical Defence Establishment at Porton to produce among other things σ_u (the standard deviation in the horizontal wind), σ_w (vertical wind), σ_T (temperature), $\overline{u'w'}$ (momentum flux) and $\overline{w'T'}$ (heat flux) for various averaging periods using a linear regression to remove long period trends.

By taking differences in variances and differences in fluxes for various averaging times (effectively a filtering technique) spectra can be derived for all the above quantities. Though individual spectra are somewhat distorted by this method average spectra, based on up to about six sets, produce results in reasonable agreement with other workers. The spectral points are given by

$$nS(n) = \sqrt{n_1 n_2} \; \frac{\Delta}{n_2 - n_1} \qquad (41)$$

where Δ is the difference between two variances or fluxes for two averaging periods and the frequencies n_1 and n_2 are the half power cut off points given by

$n_{\frac{1}{2}} = .44/s$ where s is the averaging time.

See Jones and Pasquill 1959.

Some of the individual tapes were rerun, on KDF9 at Bracknell, using the FFT. It was ensured that the same data was used for the comparison. For the results presented here the first two sets of 1024 seconds starting at 1340 GMT on the 30 October 1968 were taken. Run 1 will refer to the first 1024 seconds and run 2 to the second 1024 seconds. It was possible then to first compare regression coefficients, this being a check that the computers read the same data points, and second to compare the shapes and magnitudes of individual spectra.

Figures 1, 2 and 3 show this comparison where points derived using equation (41) are called Porton spectra. The Porton spectra used averaging times of 1,2,5,10,20,50,100,200 and 500 seconds to produce plots at geometric mean frequencies .31, .14, .06, .03, .014, .006, .003 and .0014 Hz. The plots for the individual FFT spectra were made up by averaging 32 adjacent S(n)'s giving 16 plots from each set of 512 Fourier coefficients. nS(n) was formed and plotted at the appropriate geometric mean frequency. As can be seen in figures 1(a), 2(a) and 3(a) averaging over 32 is a little too coarse at low frequencies and a little too fine at high frequencies to compare thoroughly with the Porton spectra. Nevertheless a reasonable agreement is found, with the FFT spectra giving a more consistent picture from run 1 to run 2. All these spectra are of course 'aliassed' the effect being particularly noticeable for the nS_w spectra and for this reason these spectra are not presented to demonstrate the actual form of these spectra but only to illustrate the comparison. The generally poorer agreement at low frequencies, between the individual spectra, is caused by the rapidly decreasing statistical reliability in the Porton spectra as the averaging time is increased, for example for averaging times of 500 seconds only two points finally go to evaluate the standard deviations and fluxes whereas for the FFT spectra all points have the same reliability. At higher frequencies the agreement is much better, with the Porton spectra on the whole being about 20% higher than the FFT spectra.

To make further comparisons average spectra were derived as shown in figures 1(b), 2(b) and 3(b). The Porton spectra were simply averaged in pairs. Part of the computer output for the FFT spectra included an average of each of the 512 pairs of S(n)'s. An increasing number of adjacent values were used, going respectively from low to high frequencies, to produce the average FFT spectra. For example for the lowest frequency plot, S(n)'s at frequencies 1/1024 to 5/1024 Hz were averaged together so that nS(n) was worked out and plotted at a frequency of $\sqrt{(1/1024)(5/1024)}$ i.e. at .0022 Hz. The complete set of frequencies was .0022, .0072, .0162, .0382, .104, .20, .31, and .435 Hz. It can be seen that there is fairly close agreement and even the agreement at low frequencies is not bad considering the lack of statistical stability there.

It is important to note that the FFT spectra were produced in about one tenth of the time it took to produce just the variances and fluxes for the Porton spectra. The total time, including reading in on paper tape some 7000 numbers, required to produce the final FFT spectral points as illustrated plus four other spectra, not shown, amounted to about 5 min computer time on KDF9. Though the Mercury computer is slower nevertheless to arrive at the same result would have taken over an hour. The 'averaging time technique' would probably have taken about 35 minutes on KDF9.

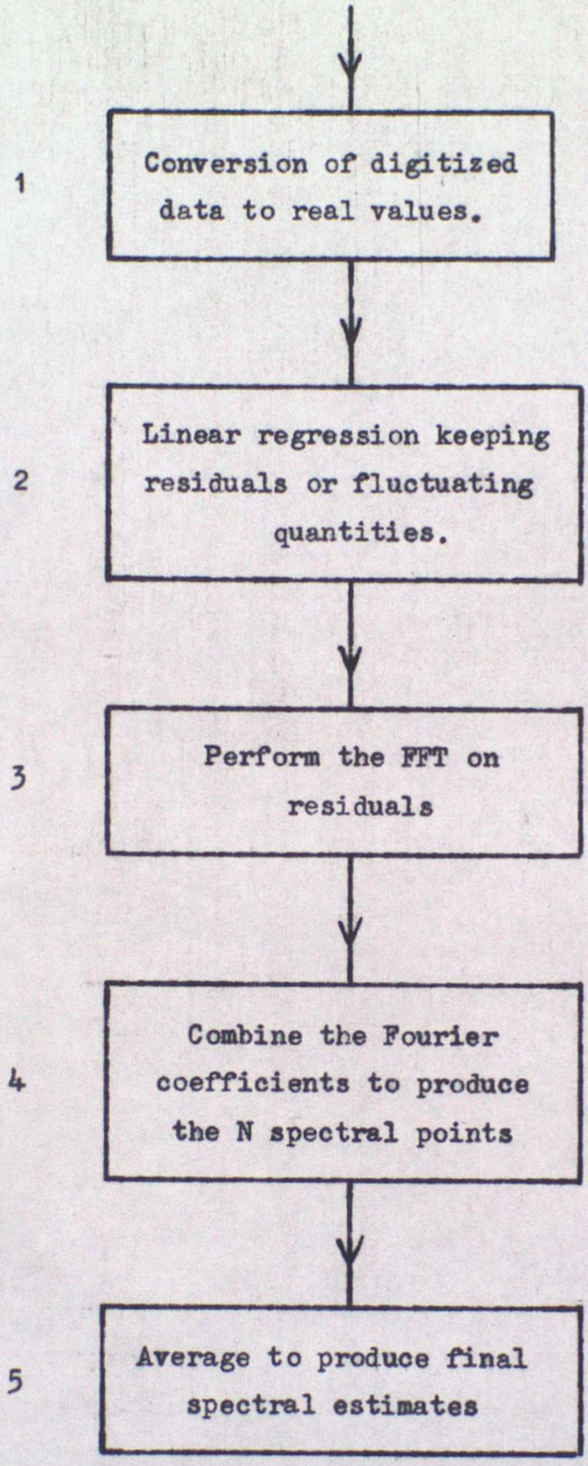

1
Conversion of digitized data to real values.
2
Linear regression keeping residuals or fluctuating quantities.
3
Perform the FFT on residuals
4
Combine the Fourier coefficients to produce the N spectral points
5
Average to produce final spectral estimates

Fig. A Flow diagram for producing spectra from digitized data

Fig. 1(b)

Average Spectra of Runs 1 and 2

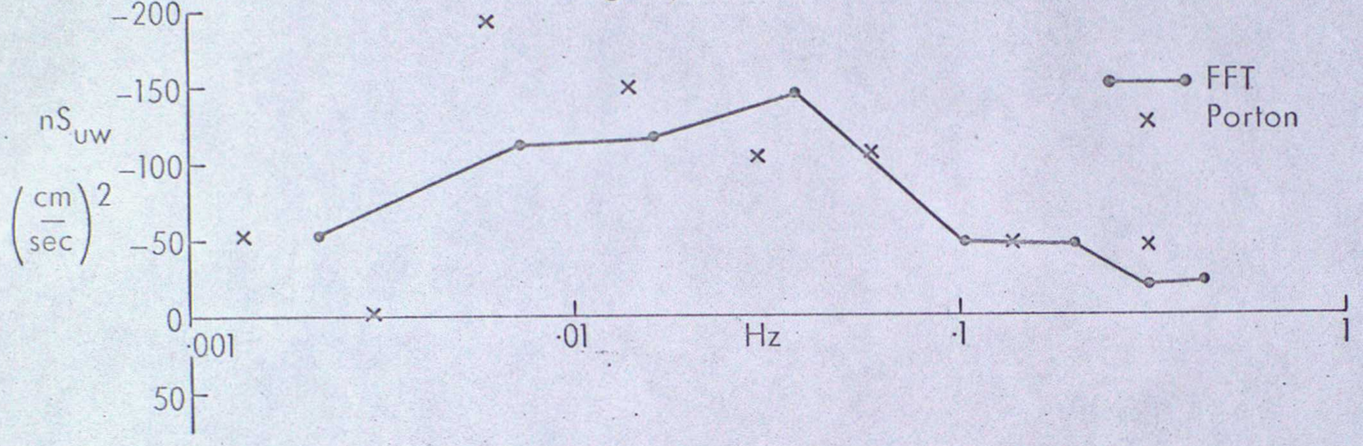


Fig. 1(a)

Individual Spectral Comparison for Runs 1 and 2

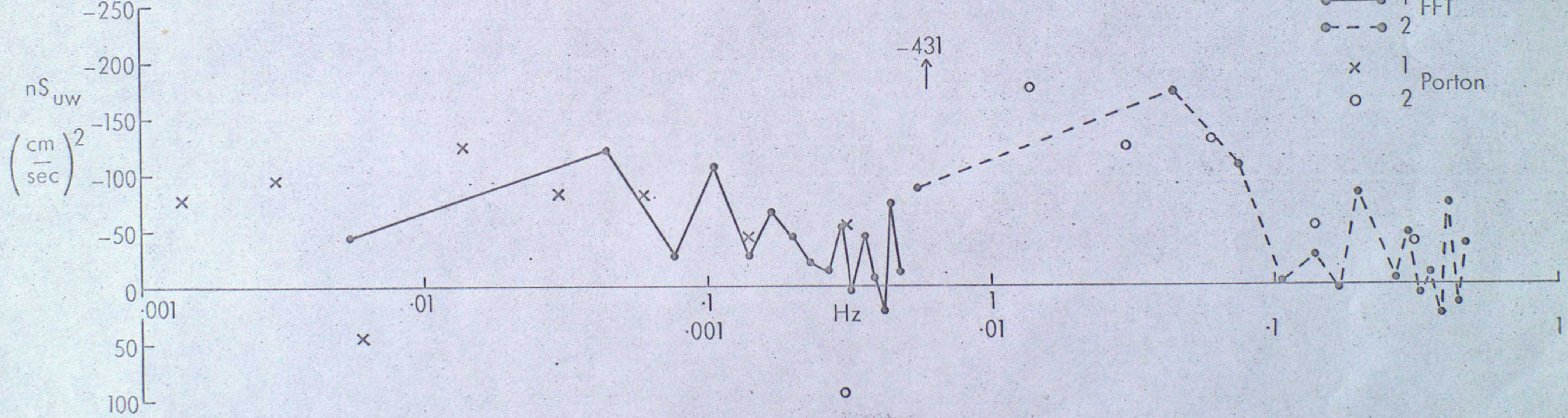


Fig. 2(b). Average Spectra of Runs 1 and 2

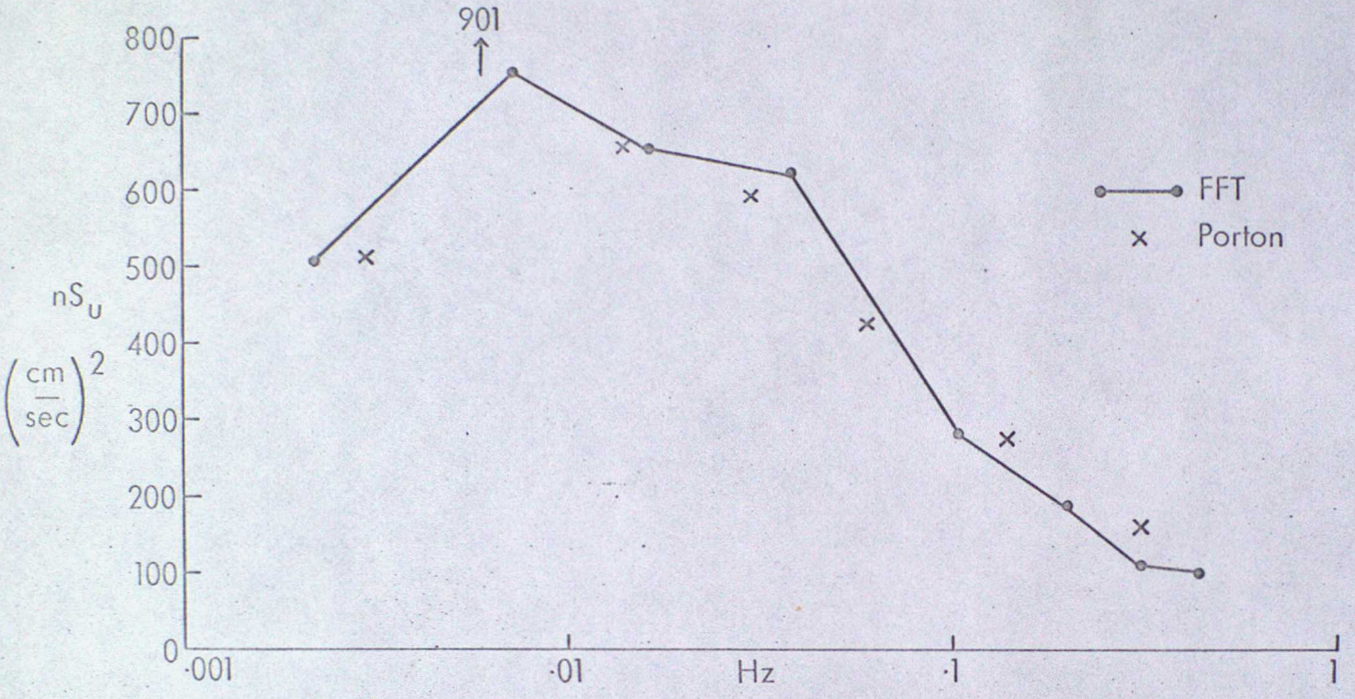


Fig. 2(a) Individual Spectral Comparison for Runs 1 and 2

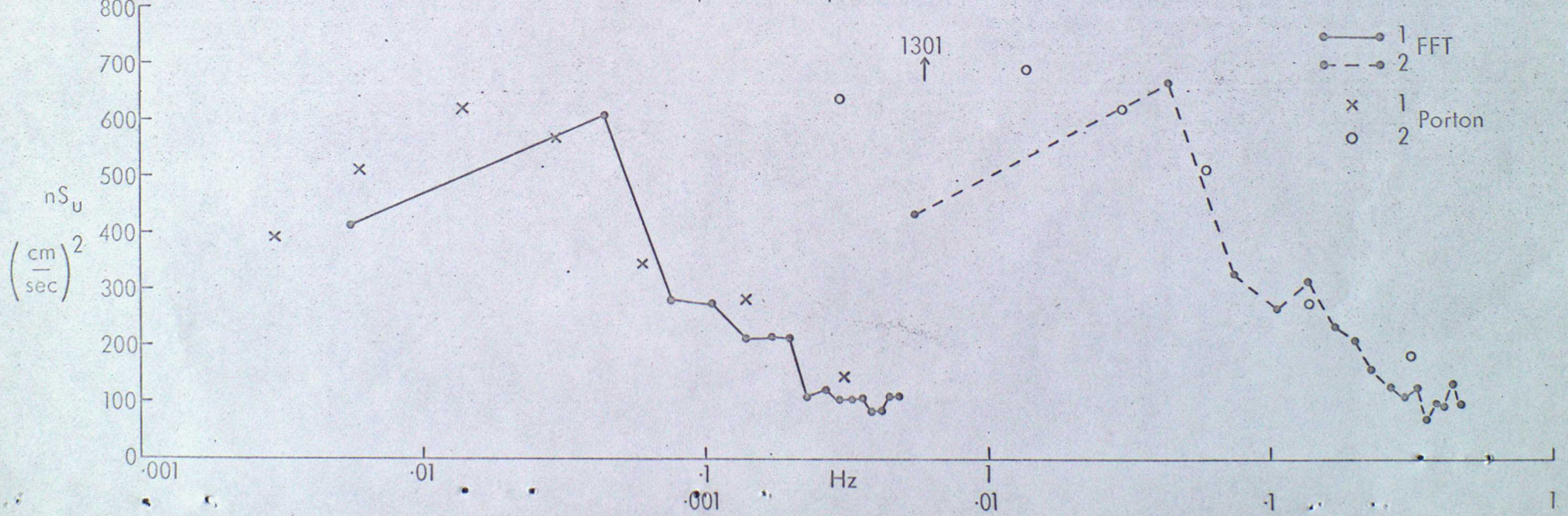


Fig. 3(b) Average Spectra of Runs 1 and 2

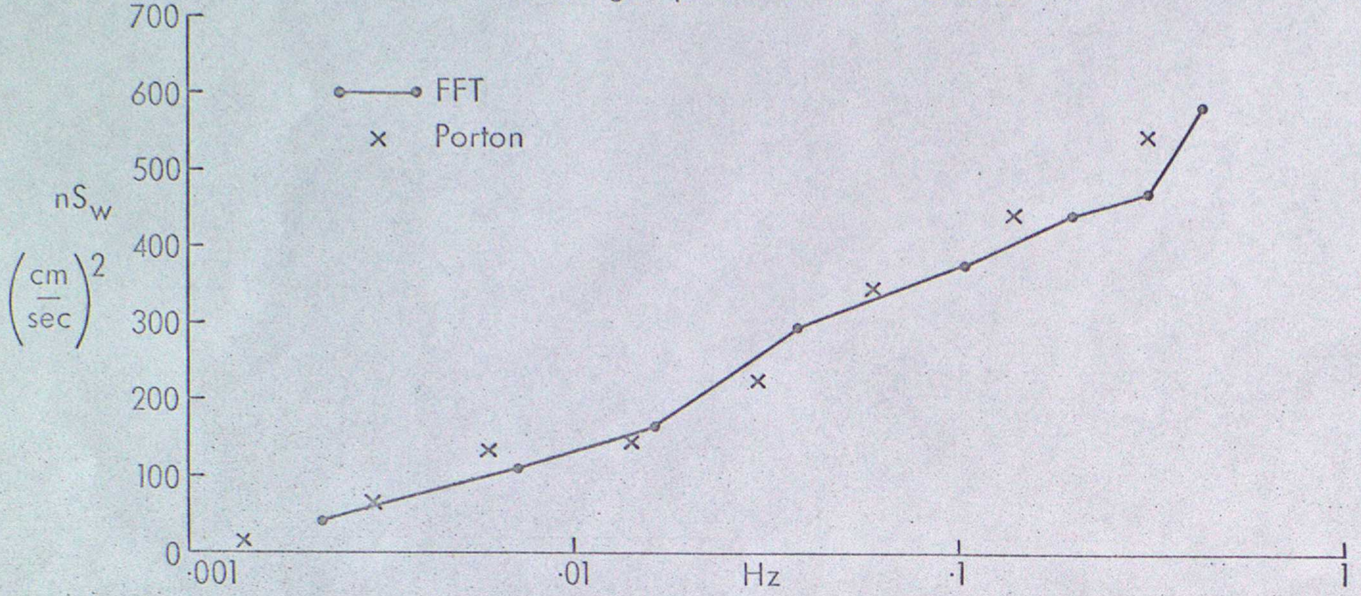
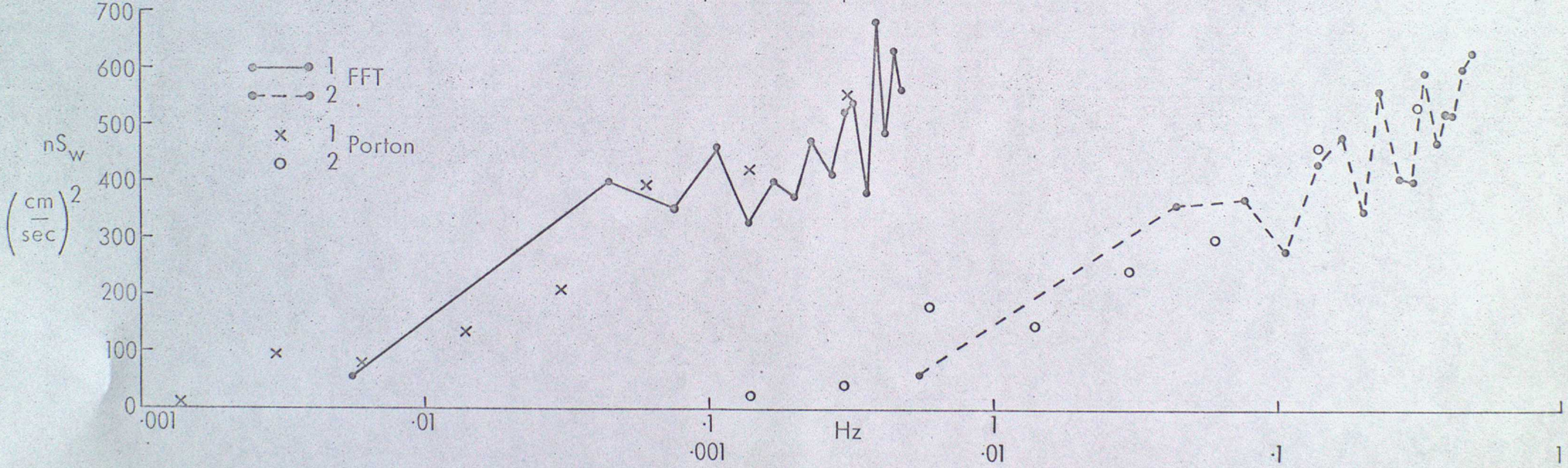


Fig. 3(a) Individual Spectral Comparison for Runs 1 and 2



APPENDIX II

A FFT computer programme

The programme is written in KDF9 User code language, and consists basically of four sub-routines, P1V1∅ and P2V2 the basic FFT programme, and two subsidiary routines P3 and P4V1 for running the FFT programme in its 'odd-even and dual' use.

The following list of preliminaries applies to P1V1∅ and P2V2:-

1. The routines evaluate Fourier coefficients given by

$$a_k = \sum_{j=o}^{N-1} x_j \cos(2\pi jk/N)$$

and

$$b_k = \sum_{j=o}^{N-1} x_j \sin(2\pi jk/N)$$

where k = 0,1,--N-1. These two can be combined to give the complex coefficient A_k where

$$A_k = \sum_{j=o}^{N-1} X_j \exp(\pm 2\pi ijk/N)$$

where X_j is a complex data point.

2. The number of data points is $N = 2^m$ real points and for 'raw' use $N = 2^m$ zeros.

3. To enter the routine the nesting store requires:-

in N1 QN/A1/A2

in N2 Qm/A3/A4

and in N3 ± 1

where A1 is the start address of real data (lengths will be N)

A2 is the start address of imaginary data (zeros)

A3 is the start address of a third store (for transferring and answers a_k)

A4 is the start address of a fourth store (for transferring and answers b_k)

Note that start addresses must be different and the stores must not overlap.

If N3 is +1 exp(+...) will be worked and

If N3 is -1 exp(-...) will be worked in the last quoted equation.

4. The main store area required apart from the programme is 4N.

5. The nesting store down to N14 is used.

6. Q7 to Q15 is used.

7. Overflow register may be used.

8. Library sub-routines L5, L1∅ and L11 are used to calculate the sines and cosines.

9. (i) The inaccurate EXITS from L1∅ and L11 occur when, after multiples of π are subtracted, the result has no bit in the ten places after the binary point. Such a result is sine or cosine of a very small number and is here counted as sin ∅ and cos ∅ i.e. ∅ in N1 and 1 in N2.

   (ii) The accuracy of L1∅ (cos) and L11 (sin) is to the 39th bit of a floating point number.

A User code flow diagram is included after the programme and after that a Flow Graph illustrating how the computation proceeds for the case of N = 8.

When the FFT programme is to be used in its 'odd-even-dual' mode, in N1, N/2 must replace N and in N2, m-1 must replace m. In addition some routine to sort the data into odd numbered and even numbered data points must be used. P3 performs this sorting and it requires:-

N1 Q N/2/A1/A2

N2 Q ± 1/-/A3

where if the counter part of N2 is +1 the first half of the data, in serial order, is to be found in the stores with a start address A1 and the second half of the data is in the stores with the start address A2. P3 then puts the N/2 even numbered data points into A1 stores and the odd numbered data points into A2 stores. N1 for P3 can thus be used as N1 for P1V1∅. If the counter part of

N2 is -1 all the data, in serial order, is to be found in stores with the start address A1. A third set of N/2 stores with a start address A3 must now be declared. P3 now gives the same result as before. Note that A3 stores will not be cleared. Once the data is sorted P1V1∅ can now be entered using the N1 of P3 for the N1 of P1V1∅. Note additionally that N3 for P1V1∅ must be +1. The coefficients hence found, are not the required Fourier coefficients, and need to be combined according to equations (36) and (37). P4V1 is a subroutine which performs this combination, the result being the desired Fourier coefficients. Entry to P4V1 requires:-

$$\text{N1} \quad \text{Q} \frac{N}{2} / \text{A1/A2}$$

$$\text{N2 Q} \quad -/\text{A3/A4}$$

where A1 is the start address for the a_k coefficients indexed a_1 to $a_{N/2-1}$ and A2 is the start address for the corresponding b_k coefficients. A3 is the start address of the stores where the N/2 real coefficients from the FFT programme can be found (in terms of equation (33) these will be indexed Re $A(0)_{xy}$ to Re $A(^{N}/2)_{xy}$) and A4 is the start address for the N/2 corresponding imaginary coefficients. Usually N1 for P4V1, will be the same as N1 for P3 and P1V1∅ and N2 for P4V1 can be the same as N2 for P1V1∅.

P3 uses the following computer facilities:-

1. The nesting store down to N3
2. Q1∅ to Q15 are used.

P4V1 uses:-

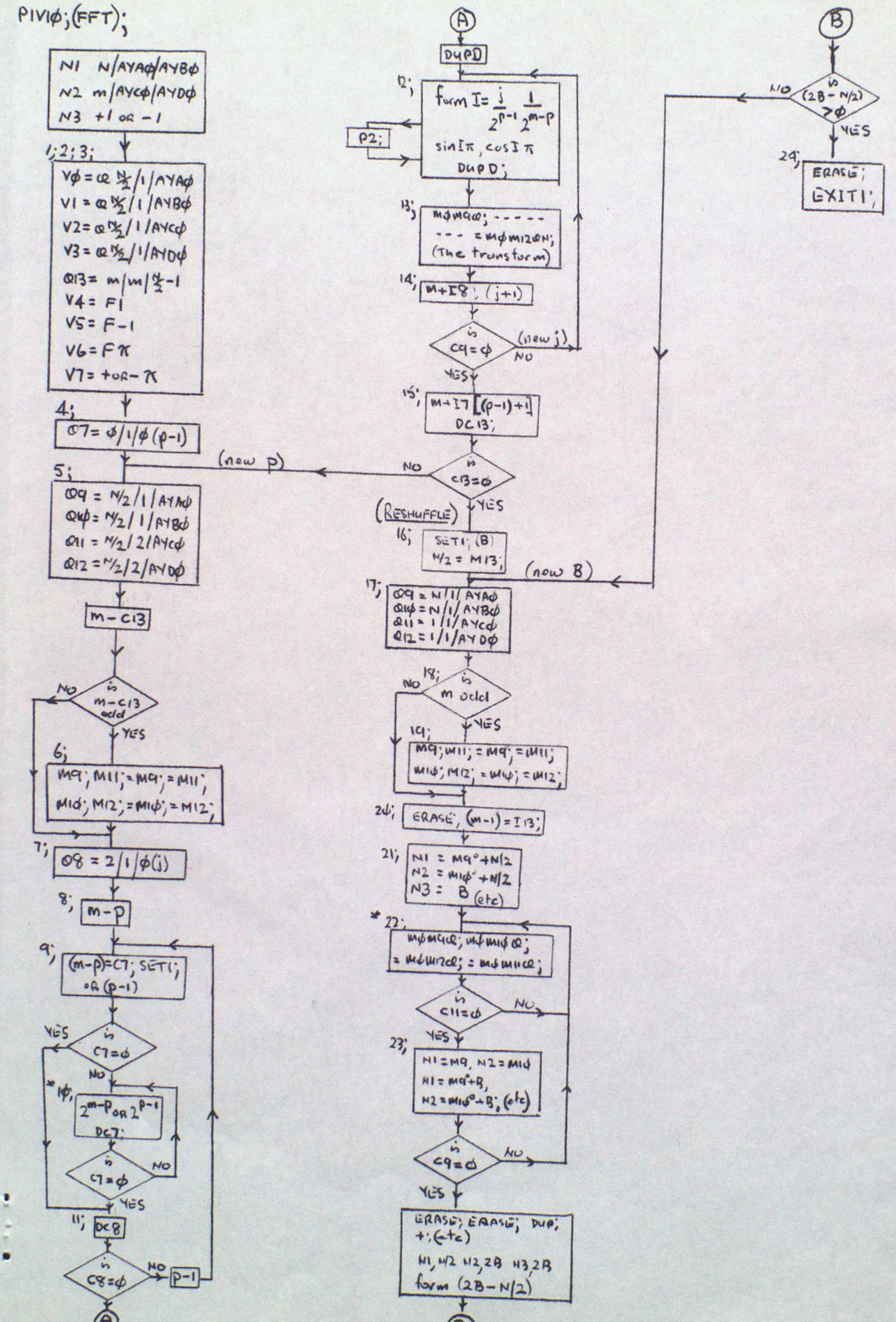
1. The nesting store down to N9
2. Q8 to Q15 are used
3. Sub-routine P2V2 is used and therefore the library subroutines and comments about those are the same as the FFT programme.

No claim is made about the particular efficiency of these subroutines but they have all been tested on the KDF9 computer.

FFT USER CODE SUBROUTINE.

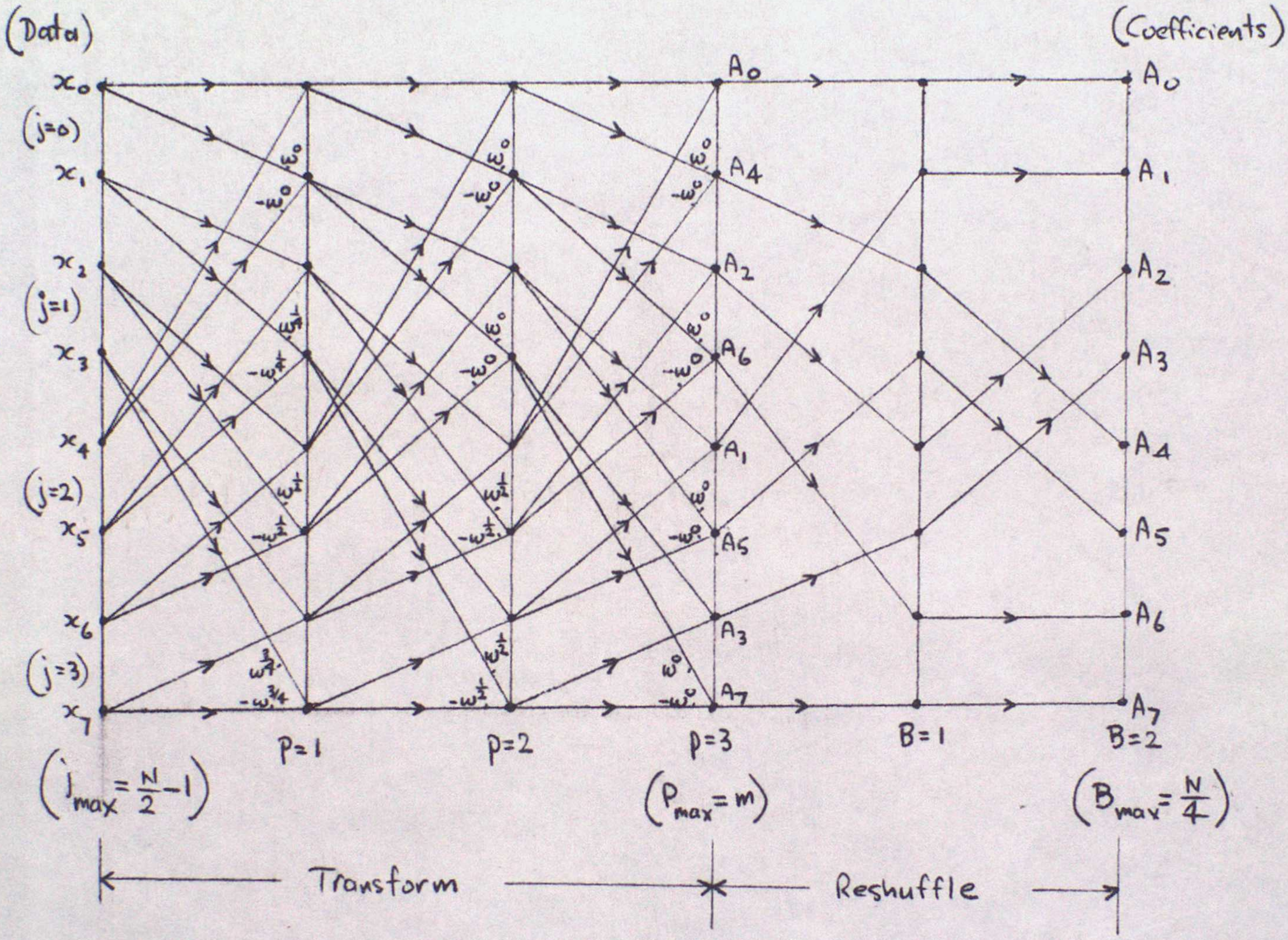
```
P1V1Ø; (FFT);
1;      DUP; =Q1Ø; =Q11; C1Ø; SET2; ÷I; ERASE; DUP; =C1Ø; =C11; I1Ø; =M1Ø;
        I1Ø=+1; I11=+1;
2;      DUP; =Q8; =Q9; I8; =M8; I8=+2; I9=+2; C1Ø; DUP; DUP; NEG; NOT; =M13;
        C8; DUP; =C13; =I13; =C8; =C9;
3;      Q1Ø; =VØ; Q11; =V1; Q8; =V2; Q9; =V3; V4=F1; V5=F-1; V6=F3·1415927;
        SET47; FLOAT; V6; xF; =V7;
4;      ZERO; =RC7;
5;      VØ; =Q9; V1; =Q1Ø; V2; =Q11; V3; =Q12; I13; C13; -; SET2; ÷I; J7=Z;
6;      M9; M11; =M9; =M11; M1Ø; M12; =M1Ø; =M12;
7;      ERASE; SET2; =RC8;
8;      I13; M7; NOT; NEG; -;
9;      =C7; SET1; J11C7Z;
*1Ø;    SET2; xD; CONT; DC7; J1ØC7NZS;
11;     DC8; J12C8Z; M7; J9;
12;     DUPD; M8; REV; ÷I; ERASE; SET47; FLOAT; REV; SET47; FLOAT; ÷F; JSP2; DUPD;
13;     MØM9Q; M9M13; DUPD; MØM1ØQ; M1ØM13; DUPD; +F; =MØM12; REVD; +F; =MØM11;
        -F; PERM; -F; PERM; DUP; PERM; xF; REVD; DUP; PERM; xF; CAB; -F; =MØM11QN;
        CAB; xF; PERM; xF; +F; =MØM12QN;
14;     M+I8; J12C9NZ; ERASE; ERASE;
15;     M+I7; DC13: J5C13NZ;
        (RESHUFFLE);
16;     SET1; M13; NOT; NEG; =M13;
17;     VØ; =Q9; V1; =Q1Ø; V2; =Q11; V3; =Q12; C9; DUP; +; =C9; C1Ø; DUP; +; =C1Ø;
        DUP; =C11; DUP; =C12; I11=+1; I12=+1;
18;     I13; SET2; ÷I; J19≠Z; J2Ø;
19;     M9; M11; =M9; =M11; M1Ø; M12; =M1Ø; =M12;
2Ø;     ERASE; I13; NEG; NOT; =I13;
21;     M1Ø; M13; DUP; PERM; +; REV; M9; +;
*22;    MØM9Q; MØM1ØQ; =MØM12Q; =MØM11Q; J22C11NZS;
23;     M9; REV; =M9; REV; M1Ø; REV; =M1Ø; REV; CAB; DUP; =C12; DUP; =C11; PERM;
        J22C9NZ; ERASE; ERASE; DUP; +; DUP; M13; -; J24> Z; J17;
24;     ERASE; EXIT1;
P2V2; (SINCOS 1);
1;      DUP; J3=Z; DUP; V4P1; -F; J4=Z; V7P1; xF; DUP;
2;      JSL1Ø; J3; REV; JSL11; J3; J5;
3;      ERASE; V4P1; ZERO; J5;
4;      ERASE; V5P1; ZERO; J5;
5;      EXIT1;
```

Flow diagram for the FFT programme.



Flow graph for N = 8 points, illustrating the FFT conversion of data to Fourier coefficients.

[Convention: $\xrightarrow{}$ means +, $\xrightarrow{W}$ means $+(\times W)$.]



1/ $w = e^{\pm\pi i}$ and the w index is $(j/\,2^{p-1})_I\ (1/2^{m-p})$.

2/ Each section transfers back and forth between main store areas.

3/ B is an index (batch number) and is equal to N/4.

4/ The transform programme is partly based on a method by R.C.Singleton (June 1967).

Subroutines to be used with the FFT programme for 'odd-even dual' use.

```
P3; (OE SORT);
1;      =Q15; =Q12; C12; DUP;
2;      Q15TOQ14; I14=+1; Q14TOQ11; Q15TOQ13; I13=+1; I15; =M13; J4< Z;
        C13; SET2; ÷I; ERASE; =C13; I12=+1; C14TOQ12; Q12TOQ15;
*3;     MØM14Q; =MØM12Q; *J3C12NZ;
4;      ZERO; =RC1Ø; Q11TOQ14;
*5;     M13M1Ø; M13M1ØQN; =MØM14Q; =MØM13Q; J5C13NZS; J7< Z;
        Q15TOQ12; ZERO; =RC1Ø; I1Ø=+2;
*6;     M12M1Ø; M12M1ØQN; =MØM14Q; =MØM13Q; J6C14NZS;
7;      EXIT1;


P4V1; (OE COM);
        =Q8; =Q9; C8; DUP; SET47; FLOAT; =VØ; SET2; SET47; FLOAT; =V1;
1;      NEG; NOT; DUP; =M13; SET-1; =I13; SET1; =RM12; =C12; I8; =RM1Ø;
        M8; =RM11; I9; =RM8; I9=+1;
2;      M8M12; M9M12Q; M8M13; M9M13Q;
3;      CAB; DUPD; REV; -F; PERM; +F; REVD; DUPD; +F; PERM; REV; -F; CAB; DUPD;
4;      M12; SET1; -; SET47; FLOAT; VØ; ÷F; JSP2;
5;      REV; DUP; REVD; PERM; xF; CAB; REVD; DUP; CAB; xF; PERM; REVD; -F; CAB;
        REVD; xF; CAB; REVD; xF; CAB; +F; PERM; +F; PERM; +F; V1; DUP; PERM; ÷F;
        PERM; ÷F; =MØM1ØQ; =MØM11Q; J2C12NZ; EXIT1;
```

REFERENCES

1. BINGHAM, C., et al, "Modern Techniques of Power Spectrum Estimation".

2. COCHRAN, W. T., et al, "What is the Fast Fourier Transform".

3. SINGLETON, R. C., "A Method of Computing the Fast Fourier Transform with Auxiliary Memory and Limited High-Speed Storage".

4. SINGLETON, R. C. and POULTER, T. C., "Spectral Analysis of the Male Killer Whale".

The above four and other useful papers on the FFT are given in the IEEE Transactions on Audio and Electroacoustics, Vol. AU-15, No. 2, June 1967.

5. GENTLEMAN, W. M. and SANDE, G., "Fast Fourier transforms for fun and profit", AFIPS proceedings. Fall Joint Computer Conference. Washington, 1966.

6. BLACKMAN, R. B. and TUKEY, J. W., "The Measurement of Power Spectra", New York, Dover, 1958.

7. COOLEY, J. W. and TUKEY, J. W., "An Algorithm for the Machine Calculation of Complex Fourier Series", Math. of Comput., Vol. 19, pp. 297-301, April, 1965.

8. JONES, R. A., "Studies of Eddy Structure in the First Few Thousand Feet of the Atmosphere", Part 3, Porton Technical Paper No. 635, April 1958.

9. RICHARDS, P. I., "Computing Reliable Power Spectra", IEEE Spectrum, Jan. 1967.

10. TUKEY, J. W., "Spectrum Calculations in the New World of the Fast Fourier Transform", Souvenir copy of a paper presented at the Advanced Seminar on Spectral Analysis of Time Series at the Army Mathematics Research Centre and Statistics Department, University of Wisconsin, Madison, Wisconsin, Oct. 1966.

11. JONES, J. I. P. and PASQUILL, F., "An experimental system for directly recording statistics of the intensity of atmospheric turbulence", Quart. J.R. Met. Soc., Vol. 85, No. 365, July 1959.