Bolm Inst. oceanogr. S Paulo, 20 (2): 79-104, 1971

A HYBRID ALGORITHM FOR THE RAPID FOURIER TRANSFORM OF EXTENSIVE SERIES OF DATA

(Received 7/6/1971)

A. S. FRANCO & N. J. ROCK
Instituto Oceanográfico da Universidade de São Paulo

SYNOPSIS

A technique is described for the rapid Fourier transform of large series of numbers. The technique takes advantage of the fact that most digital series are highly factorizable by the number 2, which permits the use of the F.F.T. algorithm.

Using two magnetic tape units, or alternatively magnetic disk facilities, very large series can be transformed efficiently with only modest computer facilities.

For the transformation of odd-valued series the Thomas Prime-Factor and Gentleman and Sande algorithms are treated in detail.

1 - GENERAL SURVEY ON FOURIER ANALYSIS

The Fourier transform has long been known to scientists for its usefulness in representing a variety of periodic phenomena. For continous signals the transform pair may be written:

$$G(f) = \int_{-\infty}^{\infty} g(t)e^{-i2\pi ft}dt \qquad (1a)$$

$$g(t) = \int_{-\infty}^{\infty} G(f)e^{i2\pi ft}df \qquad (1b)$$

for a frequency f and time t without finite limitations (i.e. $-\infty < t < \infty$, etc.). In the particular example cited above the Fourier transform permits the representation of a function of time (g(t)) by a function in the frequency domain (G(f)) and vice-versa; hence the name of transform. It serves equally to transform other domains such as wave-number and horizontal space.

PUBL. Nº 336 DO INST. OCEAN. DA USP.

For earth scientists the obtention of continuous signals in machine-processable form is usually prohibitively expensive and, for the majority of applications, unnecessary. Consequently the continuous signal is usually sampled at equal intervals of space or time, in which case a different form of the Fourier transform is used, that is applicable to the discrete values obtained by sampling. Called the Discrete Fourier Transform, or D.F.T., the transform pair may be written, for N-valued series:

$$X(k) = (1/N) \sum_{n=0}^{N-1} c(n) e^{-i2\pi kn/N} \qquad (k=0,1,2,\dots N-1) \qquad (1c)$$

$$c(n) = \sum_{k=0}^{N-1} X(k) e^{i2\pi nk/N} \qquad (n=0,1,2,\dots N-1) \qquad (1d)$$

An examination of the indexing shows that the number of mathematical operations required to evaluate the D.F.T. of an N-valued series is proportional to N^2. Consequently, for very large series the time and expense to evaluate a D.F.T. become prohibitive. This, coupled with the fact that by itself the D.F.T. has little significance in representing the signal of a random "noise", caused the method to be overshadowed by other analytical techniques with better computational speeds (e.g. convolution and spectral analysis).

The D.F.T. had all the making of a mathematical dinosaur, when Cooley and Tukey (1965) showed that a remarkable increase in computational speed can be achieved if N is a highly factorizable number. Thus if $N=r^m$, the D.F.T. can be broken down into r separate D.F.T.'s of size r^{m-2} etc.. Finally, one arrives at an m-step algorithm, each step of which requires N.r operations.

It can be seen that the number of operations has been reduced substantially by a factor of r.m/N (viz: N^2 versus N.r.m).

Franco (1970) has shown the process of the sub-division of the larger D.F.T.'s into smaller ones for the case where $N=2^m$.

For binary digital computers the case where $N=2^m$ has important advantage over other factors of r, both for multiplication economy and in addressing. Accordingly the algorithms derived for $N=2^m$ both by Cooley and Tukey, and Cooley and Sande, have acquired the name of the Fast Fourier Transform or F.F.T.

The high computational speed of the F.F.T., has made it not only feasible but also economically attractive, in terms of computer costs, to calculate energy spectra and correlation functions via the F.F.T.

Franco and Rock (1971) have demonstrated the suitability of the F.F.T. for the harmonic analysis of tides, where $N=2^{13}=8192$ hourly observations of tidal heights, or nearly a years data. By the use of matrices, tidal components centred at frequencies other than exact harmonics of the fundamental frequency were successfully extracted. Further filtering produced both the tidal and the residual energy spectrum.

However the fact that the analysis is tied to a power of 2 is a serious drawback for many people. Few scientists are willing to neglect available data and frequently better conditioned matrices, and better rejection of interfering components may be obtained by an astute choice of N. Fortunately, if we are willing to sacrifice computational speed, a more generalized form of the F.F.T. may be derived.

2 - THE GENTLEMAN AND SANDE ALGORITHM

Let us consider the simplified case where N has a factor *p*, and having obtained the D.F.T.'s of *p* separate series each of N/p numbers, we now require the algorithm to combine *p* sets of N/p Fourier coefficients.

We may write the original D.F.T. as:

$$c(n) = \sum_{k=0}^{N-1} X(k)e^{-2\pi nk/N} \qquad (n=0,1,2,\ldots N-1) \qquad (2a)$$

or using the notation:

$$W_N = e^{-2\pi i/N} \qquad (2b)$$

$$c(n) = \sum_{k=0}^{N-1} X(k)W_N^{nk} \qquad (n=0,1,2,\ldots N-1) \qquad (2c)$$

Now let k take the form $k=(bp+j)$ $\begin{cases} j=0,1,2,\ldots p-1 \\ b=0,1,2,\ldots(N/p)-1 \end{cases}$

and let n take the form $n=(a+m(N/p))$ $\begin{cases} m=0,1,2,\ldots p-1 \\ a=0,1,2,\ldots(N/p)-1 \end{cases}$

We may now rewrite (2*a*) in terms of two separate summations:

$$c(a+m(N/p)) = \sum_{j=0}^{p-1} \sum_{b=0}^{(N/p)-1} X(bp+j)W_N^{(bp+j)(a+m(N/p))} \qquad (2d)$$

Note that if we write the original series X(k) as a two dimensional p.(N/p) array, the rearrangement of the data in equation (2*d*) corresponds to a row-wise indexing instead of the column-wise indexing normally used in digital computers.

Working equation (2*d*) through, we get:

$$c(a+m(N/p)) = \sum_{j=0}^{p-1} \sum_{b=0}^{(N/p)-1} X(bp+j)W_N^{(bpa+j(a+m(N/p))+mbN)}$$

$$= \sum_{j=0}^{p-1} \sum_{b=0}^{(N/p)-1} X(bp+j)W_N^{bpa}.W_N^{j(a+m(N/p))}.W_N^{mbN}$$

but, according to (2*b*),

$$W_N^N = 1 \qquad \text{and} \qquad W_N^p = e^{-2\pi i/(N/p)}W_{(N/p)}$$

Thus rearranging

$$c(a+m(N/p)) = \sum_{j=0}^{p-1} W_N^{j(a+m(N/p))} \sum_{b=0}^{(N/p)-1} X(bp+j)W_{N/p}^{ba} \qquad (2e)$$

Cross-referring between the above equation and equation (2*c*) it can be seen that the innermost summation is already in the form of an N/p valued D.F.T.

Let $$B_j(a) = \sum_{b=0}^{(N/p)-1} X(bp+j)W_{N/p}^{ba}$$

By reverting the indexing to the column-wise form, we create series, each of *p* Fourier coefficients, and

$$c(a+m(N/p)) = \sum_{j=0}^{p-1} B_a(j)W_N^{j(a+m(N/p))} \qquad (2f)$$

The complex multiplier of equation (2*e*) is more easily written:

$$W_N^{j(a+m(N/p))} = (W_N^a . W_p^m)^j$$

since according to (2*b*):

$$W_N^{N/p} = W_p$$

from which it can be seen that the normal multiplier of the D.F.T. (viz: W_p^{mj}) is multiplied by an additional corrective factor W_N^{aj}, called the "twiddle factor" by its originators Gentleman and Sande (1966), which serves to shift the complex coefficients cyclically so that N/p and *p* may be identical or factorizable one by the another. When p=2 and N=2γ successive repetitions of the algorithm make it formally similar to the F.F.T..

3 - THOMAS PRIME-FACTOR ALGORITHM FOR TWO FACTORS

If N can be expressed by

$$N = pq \qquad (3a)$$

where *p* and *q* are prime with respect to one another, we can use this property to eliminate the "twiddle factor" by means of a suitable sequence.

As before we write:

$$c(n) = \sum_{k=0}^{N-1} X(k)W_N^{nk} \qquad (3b)$$

and put:

$$k = (jp+mq) \; Mod \; N \qquad \begin{cases} j=0,1,2\ldots q-1 \\ m=0,1,2\ldots p-1 \end{cases} \qquad (3c)$$

which defines the remainder of the interger division of (jp+mq) by N. It is possible to prove that *k* takes all the values in the interval

$$0 \leqslant k \leqslant N-1 \qquad (3d)$$

The input data can thus be arranged as p sequences of q numbers. For p=7 and q=3 an example of the two dimensional mapping of 21 numbers appears as in Table 3-I.

TABLE 3-I - k = (7p+3q) *Mod* 21

j \ m	0	1	2	3	4	5	6
0	0	3	6	9	12	15	18
1	7	10	13	16	19	1	4
2	14	17	20	2	5	8	11

Similarly we will suppose that output data sequence is represented in the form:

$$n = (gI+hJ) \; Mod \; N \qquad \begin{cases} g=0,1,2,\ldots \; p-1 \\ h=0,1,2,\ldots \; q-1 \end{cases} \tag{3e}$$

where I and J are to be determined at our convenience.

Before replacing k and n in W_N^{nk}, by expressions (3*c*) and (3*e*) it is convenient to derive a general expression for W_M^L, according to the definition of the operator *Mod*. If L is any integer so that

$$L = KM + \beta$$

where K is the quotient of the integer division of L by M, and β is the remainder of that division, we have according to (2*b*):

$$W_M^L = e^{-i2\pi(KM+\beta)/M}$$

$$= e^{-i2\pi K} e^{-i2\pi\beta/M} = W_M^{\beta} = W_M^{L \; Mod \; M} \tag{3f}$$

Consequently

$$W_N^{nk} = W_N^{(nk) \; Mod \; N}$$

and from (3*c*) and (3*e*)

$$W_N^{nk} = W_N^{\left[(jp+mq) \; Mod \; N \; (gI+hJ \; Mod \; N\right] \; Mod \; N}$$

or, according to Appendix I, formula (*d*):

$$W^{nk} = W_N^{(jp+mq)(gI+hJ)}$$

$$= W_N^{(jgpI+jhpJ+mgqI+mhqJ)}$$

But, from (2*b*) and (3*a*) we have:

$$W_N^p = e^{-i2\pi p/pq} = W_q$$

and

$$W_N^q = e^{-i2\pi q/pq} = W_p$$

thus

$$W_N^{nk} = W_q^{igI} . W_q^{jhJ} . W_p^{mgI} . W_p^{mhJ} \tag{3g}$$

Since we can choose I and J at our convenience, these factors may be chosen to satisfy the following relationships:

$$\begin{aligned} W_q^I &= W_q^{I \; Mod \; q} = W_q^0 \\ W_q^J &= W_q^{J \; Mod \; q} = W_q \\ W_p^I &= W_p^{I \; Mod \; p} = W_p \\ W_p^J &= W_p^{J \; Mod \; p} = W_p^0 \end{aligned} \tag{3h}$$

which means that I and J must be given by

$$\begin{cases} I \; Mod \; q = 0 \\ I \; Mod \; p = 1 \end{cases} \qquad \begin{cases} J \; Mod \; q = 1 \\ J \; Mod \; p = 0 \end{cases} \tag{3i}$$

thus, according to (3*h*), expression (3*g*) reduces to:

$$W_N^{nk} = W_q^{jh} . W_p^{mg} \tag{3j}$$

Consequently, by using (3*c*), (3*b*), (3*e*) and (3*j*) we can change (3*b*) into:

$$c \left[(gI+hJ) \; Mod \; N \right] = \sum_{j=0}^{q-1} \sum_{m=0}^{p-1} X \left[(jp+mq) \; Mod \; N \right] W_q^{jh} W_p^{mg}$$

or

$$c \left[(gI+hJ) \; Mod \; N \right] = \sum_{j=0}^{q-1} W_q^{jh} \sum_{m=0}^{p-1} X \left[(jp+mq) \; Mod \; N \right] W_p^{mg}$$

This expression can be split into two, by using a more suitable matrix notation:

$$||W_p^{mg}|| \; \{X_j(m)\} = \{\alpha_j(g)\}$$

and

$$||W_q^{jh}|| \; \{\alpha_g(j)\} = \{c_g(h)\}$$

Since we have q values of j and p values of g, there will be q groups of values of $\alpha_j(g)$, each one with p values. This is the result of the first step. Now since we have p values of g and q values of h the result of the second step will be p groups of values of $c_g(h)$ each one with q values. In other words we have q analyses with a $p \times p$ matrix and p analyses with a $q \times q$ matrix.

Table 3-II gives the output mapping for p=7 and q=3.

TABLE 3-II - n = (15q+7h) *Mod* 21

h \ g	0	1	2	3	4	5	6
0	0	15	9	3	18	12	6
1	7	1	16	10	4	19	13
2	14	8	2	17	11	5	20

Another possibility exists to choose I and J so that

$$\begin{cases} I \ Mod\ q = 0 \\ I \ Mod\ p = p-1 \end{cases} \qquad \begin{cases} J \ Mod\ q = q-1 \\ J \ Mod\ p = 0 \end{cases}$$

In this case it is easy to prove that the conjugates of c(n) are found, i.e., c(N-n), for n=0,1,2, ... N-1. In other words the values of n for p=7 and q=3 would be tabulated by subtracting the values of Table 3-I (except zero) from 21.

4 - THOMAS PRIME-FACTOR ALGORITHM FOR THREE FACTORS

If it is possible to split p into two mutually prime factors r and s, so that

$$p = rs \tag{4a}$$

we have

$$N = rsq \tag{4b}$$

and a new step can be added to the analysis. In fact we can make:

$$m = (ar+bs) \ Mod\ p \qquad \begin{cases} a=0,1,2,\dots s-1 \\ b=0,1,2,\dots r-1 \end{cases} \tag{4c}$$

and

$$n = (gI+hJ+lL) \; Mod \; N \qquad \begin{cases} g=0,1,2,\ldots r-1 \\ h=0,1,2,\ldots s-1 \\ l=0,1,2,\ldots q-1 \end{cases} \tag{4d}$$

where I, J and L may be chosen at our convenience.

We have from (3*c*) and (4*c*):

$$k = (jp+mq) \; Mod \; N = \{jp + \left[(ar+bs) \; Mod \; p\right]q\} \; Mod \; N$$

After some *Mod* operator algebra (see Appendix I) this expression may be changed into:

$$k = \left[j(rs) + a(rq) + b(sq)\right] \; Mod \; N \tag{4e}$$

this expression gives the input mapping. For r=3, s=5 and q=4 we have the result shown in Table 4-I.

TABLE 4-I - k = (15j + 12a + 20b) *Mod* 60
Input mapping for 3 factors = r=3, q=4, s=5

	j = 0			j = 1			j = 2			j = 3		
a \ b	0	1	2	0	1	2	0	1	2	0	1	2
0	0	20	40	15	35	55	30	50	10	45	5	25
1	12	32	52	27	47	7	42	2	22	57	17	37
2	24	44	4	39	59	19	54	14	34	9	29	49
3	36	56	16	51	11	31	6	26	46	21	41	1
4	48	8	28	3	23	43	18	38	58	33	53	13

TABLE 4-II - n = (40g + 36h + 45l) *Mod* 60
Output mapping for 3 factors = r=3, q=4, s=5

	g = 0				g = 1				g = 2			
h \ l	0	1	2	3	0	1	2	3	0	1	2	3
0	0	45	30	15	40	25	10	55	20	5	50	35
1	36	21	6	51	16	1	46	31	56	41	26	11
2	12	57	42	27	52	37	22	7	32	17	2	47
3	48	33	18	3	28	13	58	43	8	53	38	23
4	24	9	54	39	4	49	34	19	44	29	14	59

Now from (4*d*) and (4*e*) we can obtain:

$$W_N^{nk} = W_N\left[j(rs) + a(rq) + b(sq)\right]gI+hJ+lL)$$

$$= (W_N^{rsI})^{jg}.(W_N^{rsJ})^{jh}.(W_N^{rsL})^{jl}$$

$$\times\ (W_N^{rqI})^{ag}.(W_N^{rqJ})^{ah}.(W_N^{rqL})^{al}$$

$$\times\ (W_N^{sqI})^{bg}.(W_N^{sqJ})^{bh}.(W_N^{sqL})^{bl} \qquad (4f)$$

But we have from (2*b*) and (4*b*):

$$W_N^{rs} = e^{-i2\pi rs/rsq} = W_q$$

$$W_N^{rq} = e^{-i2\pi rq/rsq} = W_s$$

$$W_N^{sq} = e^{-i2\pi sq/rsq} = W_r$$

thus

$$W_N^{nk} = (W_q^I)^{jg}(W_q^J)^{jh}(W_q^L)^{jl}$$

$$\times\ (W_s^I)^{ag}(W_s^J)^{ah}(W_s^L)^{al}$$

$$\times\ (W_r^I)^{bg}(W_r^J)^{bh}(W_r^L)^{bl}$$

Since we can choose I,J and L so that

$$\left\{\begin{array}{l} I\ Mod\ q = 0 \\ I\ Mod\ s = 0 \\ I\ Mod\ r = 1 \end{array}\right. \equiv I\ Mod\ qs = 0 \qquad \left\{\begin{array}{l} J\ Mod\ q = 0 \\ J\ Mod\ r = 0 \\ J\ Mod\ s = 1 \end{array}\right. \equiv J\ Mod\ qr = 0$$

$$\left\{\begin{array}{l} L\ Mod\ s = 0 \\ L\ Mod\ r = 0 \\ L\ Mod\ q = 1 \end{array}\right. \equiv L\ Mod\ rs = 0 \qquad (4g)$$

it follows that

$$W_N^{nk} = W_q^{jl}.W_s^{ah}.W_r^{bg} \qquad (4h)$$

thus from (3*b*), (4*d*), (4*e*) and (4*h*) we obtain:

$$c\left|(gI+hJ+lL)\ Mod\ N\right| = \sum_{j=0}^{q-1}\ \sum_{a=0}^{s-1}\ \sum_{b=0}^{r-1} X\{\left[j(rs) + a(rq) + b(sq)\right]\ Mod\ N\}W_q^{jl}W_s^{ah}W_r^{bg}$$

or, by using a more suitable notation,

$$c_{gh}(l) = \sum_{j=0}^{q-1} W_q^{jl} \sum_{a=0}^{s-1} W_s^{ah} \sum_{b=0}^{r-1} X_{ja}(b)W_r^{bg}$$

By using matrix notation this expression can be split into the following formulae:

$$\| W_r^{bg} \| \{X_{ja}(b)\} = \{\alpha_{ja}(g)\}$$

$$\| W_s^{ah} \| \{\alpha_{jg}(a)\} = \{\gamma_{jg}(h)\} \qquad (4i)$$

$$\| W_q^{jl} \| \{\gamma_{gh}(j)\} = \{c_{gh}(l)\}$$

If one of the factors is a power of 2, then the respective summation can be treated by the F.F.T..

It may be noted that $X_{ja}(b)$ represents the values of X(k) arranged according to expression (*4e*) as input data; whereas $c_{gh}(l)$ are the values of c(n) appearing in the output according to the order given by (*4d*). (See Tables 4-I and 4-II for a three dimensional mapping of input and output).

5 - APPLICATION TO TIDAL SPAN

The main objection to the method of tidal analysis via F.F.T. is that the number of samples must be a power of 2. In fact Cartwright (personal communication) says that the inter-tidal bands are contaminated by tidal side bands which make it difficult to obtain the noise level without complicated corrections. Thus it may be useful to establish tidal spans which can be treated by the method here described. This can be done by choosing the number of days so that the constituents M_2, S_2, K_1 and O_1 accomplish approximately a whole number of cycles. Since we are not obliged to work with a whole number of days we have used half a day every time a better approximation could be made.

TABLE 5-I — Tidal series

Span in days	Span in hours	Factors	Number of cycles per series			
			M_2	S_2	K_1	O_1
15.0	360	5x9x8	28.984	30	15.041	13.943
29.0	696	29x3x8	56.036	58	29.079	26.957
58.0	1392	29x3x16	112.072	116	58.159	53.913
87.0	2088	29x9x8	168.108	174	87.238	80.870
104.5	2508	33x19x4	201.923	208	104.786	97.136
133.5	3204	29x27x4	257.959	267	133.866	124.093
162.5	3900	39x5x4	313.994	335	162.945	151.050
177.5	4260	71x15x4	342.979	355	177.986	164.993
192.5	4620	35x33x4	371.963	385	193.027	178.936
220.5	5292	49x27x4	460.066	441	221.103	204.963
235.5	5652	157x3x4	455.050	471	236.145	218.906
279.5	6708	43x39x4	540.070	559	280.265	259.805
297.0	7128	31x11x8	537.885	594	297.813	276.072
325.0	7800	39x25x8	627.929	650	325.890	302.099
355.0	8520	71x15x8	685.957	710	355.972	329.985
369.0	8856	41x27x8	713.009	718	370.010	342.999

Table 5-I shows the figures in days and hours. The hours have been factorized and the result is shown in the third column. Note that the last factor is always a power of 2 which means that one of the steps can be worked out by the Cooley-Tukey algorithm.

6 - NOTES ON COMPUTATION

In both the Thomas prime-factor and the Gentleman and Sande algorithm, the smallest blocks of D.F.T.'s are formed using a very efficient algorithm due to Watt (1959). The algorithm is a recurrence formula that requires only one sine and cosine to evaluate a pair of Fourier coefficients.

Briefly for an argument $\theta = 2\pi n/N$ $(n=0,1,2,\ldots N-1)$, where N is the extent of the series, the recurrence formula is

$$X_k = Y(k) + 2\cos\theta . X_{k+1} - X_{k+2} \qquad (6a)$$

Y(k) being a real-valued series sampled at equidistant intervals $(k=0,1,2,\ldots N-1)$.

Putting $X_N=0$ and $X_{N+1}=0$, the formula is iterated N times and the n^{th} harmonic Fourier coefficients found from

$$a_n = (X_o - X_1 \cos\theta)2/N$$

$$b_n = (X_1 \sin\theta)2/N$$

A fuller treatment of the method may be found in Cartwright and Catton (1963).

The chief advantage of the Thomas prime-factor algorithm is the computational speed gained by avoiding the use of the "twiddle-factor" of the Gentleman and Sande algorithm. It suffers in being restricted to only mutually prime factors, being messy and bulky to program, and the need for extra memory space to sort the output.

There exist at least two distinctive ways of programming the "twiddle factor" into the Gentleman and Sande algorithm. In equation (2*f*) for each of the N/p series $B_a(j) = (a=0,1,2,\ldots(N/p-1))$ the argument *a* of the "twiddle factor $W_{N_j}^{ja}$ is equivalent to a phase-shifting of the complex multiplier $W_p^{jm} = W_N^{jmN/p}$ which necessitates a recalculation of the multiplier $W_N^{j(a+m(N/p))}$ for each series. However the method is extremely compact to program (see Appendix II) and, by sacrificing some of the speed of the calculation, can be programmed so that the Fourier coefficients are calculated "in-place", or in other words only one array is needed to store the data at any phase of the computation.

On the other hand, to avoid continuous recalculation of the "twiddle factor", if several sets of data will use the algorithm on the same computer pass, the complex array W^{ja} can be calculated at the start of the program, stored, and multiplied directly with the coefficients $B_a(j)$.

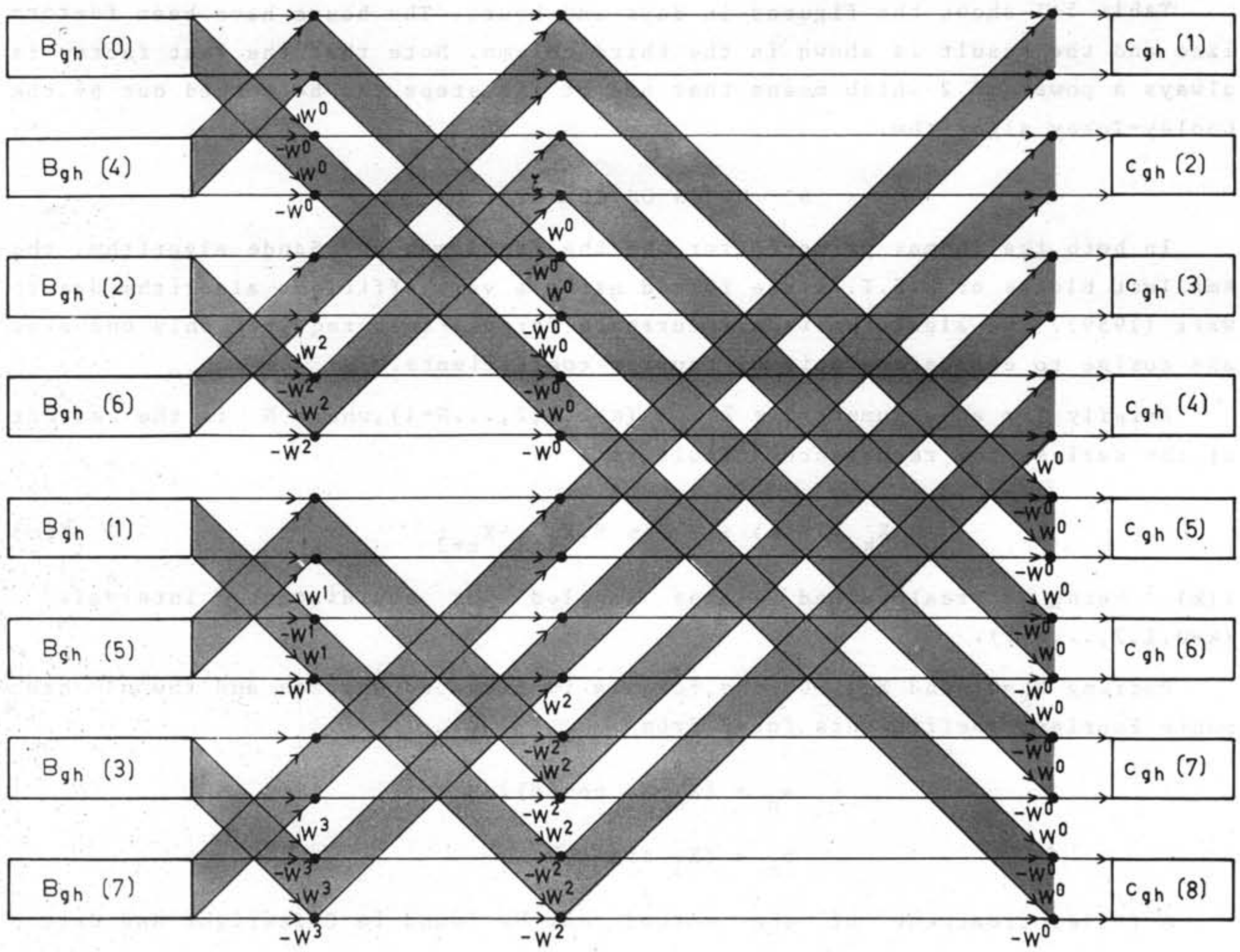


FIG. 1 — Signal flow - diagram for the F.F.T. treatment of $q=2^3$ blocks of D.F.T. s size $r \times s$.

The Watt sub-algorithm has been used as the basic building block of both the Gentleman and Sande and the Thomas prime-factor algorithm. But the F.F.T., by a series of linear combinations, avoids the use of the sub-algorithm, and can substitute the Watt algorithm with greater computational efficiency. Moreover, most series have a factor that is a power of 2 (especially time-series due to the natural divisions of days, hours, minutes and seconds), which suggests that a general purpose algorithm should take advantage of the F.F.T..

The use of the Thomas prime-factor algorithm in connection with the F.F.T. is indicated; since with one factor even and the others -of necessity-odd, there is a good possibility of finding at least three mutually prime factors. An examination of equation (4*i*) shows that if $q=2^m$ the D.F.T. consists of $s \times q$ separate calculations of an r-size D. F. T., $r \times q$ calculations on an s-size D. F. T., and $r \times s$ calculations of a $q=2^m$ size F. F. T. Leaving the F. F. T. stage until the last summation, allows combination via the F. F. T. to be effected in $q=2^m$ blocks of size $r \times s$. Note that this is identically equal to $r \times s$ blocks of q-size of F.F.T., but in the computer the former has computational advantages of speed and memory space. Figure 1 shows the treatment of the $q=2^m$ blocks by the F.F.T. algorithm, as a signal flow diagram (Cochran *et al.*, 1967). Since the F.F.T. only requires two blocks of data in the memory core at one time, the blocks of data for each stage of the calculation can be stored either on magnetic disk or magnetic tape.

The reader is referred to Cochran *et al.* (1967) and Franco (1970) for a fuller discussion of the F.F.T. algorithm.

For reasons of convenience in the manipulation of magnetic tape, a particular form of the F.F.T. was selected, where the data enters in "bit-reversed"* order and the Fourier coefficients exit in natural order. A very simple technique for calculating bit-reversed numbers is presented in Table 6-I (E. Bergamini, 1968, personal communication).

TABLE 6-I – Generation of bit-reversed series

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2^1 ↓	0	1														
2^2 ↓	0	2	1	3												
2^3 ↓	0	4	2	6	1	5	3	7								
2^4 ↓	0	8	4	12	2	10	6	14	1	9	55	13	3	11	7	15

NOTE: Each sucessive line is generated by doubling the sequence of the line above. The odd sequence to the right of the dotted separator is formed by adding one to the even sequence to the left of the separator.

Instead of arranging the blocks of data on magnetic tape in complete bit-reversed order, the odd numbers are interposed with the even numbers of sequence (viz: for a normal bit-reversed sequence 0,4,2,6,1,5,3,7, the order becomes 0,1,4,5,2,3,6,7). Initially, every other block (i.e. even numbered) is read from the first tape to form pairs for combination. The resulting pair of blocks after combination are written in sequence onto a second tape. On completion of the first half of the pass, the tape being read is rewound and the process continued reading and combining those data blocks that were skipped on the first half of the pass (i.e. odd numbers). Both tapes are then rewound, their rôles reversed and the process repeated for the second pass. For $r=2^m$ the process has *m* such iterative stages.

7 - CONCLUSION

A very large data series that is highly factorizable by 2 can thus be Fourier transformed very efficiently using very little computer memory core.

---

* By a bit-reversed number, one understands a number that when represented in binary notation has its binary bits arranged in reverse sequence to that of its natural equivalent.

Problems arise from the bit-reversing of the data at the input and sorting the out-put, both of which need additional memory core. Notwithstanding, these problems can be overcome either by the use of separate subprograms, or the extensive use of magnetic disk. The optimum solution depends on the computer configuration.

Although it appears feasible to program also the Gentleman and Sande algorithm in conjunction with the F.F.T., there are no distinct advantages in doing so and only in exceptional circunstances might programming effort be justifiable.

RESUMO

Apresenta-se neste trabalho uma técnica de transformação rápida de Fourier aplicada a uma longa série de valores numéricos. A técnica tira partido do fato de que a grande maioria das séries digitalizadas é, em geral, suscetível de fatoração onde aparece frequentemente o fator 2, o que permite o emprego do algorítmo da transformação rápida de Fourier (F.F.T.).

Com o emprego de duas fitas magnéticas ou discos, pode ser efetuada eficientemente a transformação de longas séries em computadores de modesta memória.

O algorítmo de fatores primos de Thomas e o de Gentleman e Sande são, respectivamente, tratados em detalhe, na transformação de séries com número impar de valores.

ACKNOWLEDGEMENTS

The authors gratefully wish to acknowledge the assistance of the Instituto de Física da Universidade de São Paulo in permitting extensive use of the IBM 360/44 computer.

The research was carried out whilst the junior author was a visiting professor to the Instituto Oceanográfico with the assistance of the "Conselho Nacional de Pesquisas" and the "Fundação de Amparo à Pesquisa do Estado de São Paulo".

REFERENCES

CARTWRIGHT, D. E. & CATTON, D. B. 1963. On the Fourier analysis of tidal observations. Int. hydrogr. Rev., vol. 40, no. 1, p. 113-125.

COCHRAN, W. T., COOLEY, J. W. *et al.*. 1967. What is the Fast Fourier Transform? I.E.E.E. Trans. Audio Electroacoustics, vol. AU-15, p. 45-55.

COOLEY, J. W. & TUKEY, J. W. 1965. An algorithm for the machine calculation of complex Fourier series. Maths Comput., vol. 19, p. 297-301.

DOODSON, A. T. 1928. The analysis of tidal observations. Phil. Trans. R. Soc., Ser. A, vol. 227, p. 223-279.

FRANCO, A. S. 1970. **Fundamentals of power spectrum analysis as applied to discrete observations.** Int. hydrogr. Rev., vol. 47, no. 1, p. 91-112.

________ & ROCK, N. J. 1971. The Fast Fourier Transform and its application to tidal oscillations. Bolm Inst. oceanogr. S Paulo, vol. 20, fasc. 1, p. 155-199.

GENTLEMAN, W. M. & SANDE, G. 1966. Fast Fourier Transform for fun and profit. Fall Joint Computer Conference, 1966. AFIPS Proc., vol. 29.

WATT, J. M. 1959. A note on the evaluation of trigronometric series. Comput. J., 1.4, p. 162 *Apud* CARTWRIGHT, D. E. & CATTON, D. B. 1963. On the Fourier analysis of tidal observations. Int. hydrogr. Rev., vol. XL, no. 1, p. 113.125.

APPENDIX I

It is necessary to derive some general expression involving the operator "*Mod*" in order to simplify expression (4*f*).

From the definition itself of A *Mod* N, it follows that

$$((((A \ Mod \ N) \ Mod \ N) \ Mod \ N \ldots))) = A \ Mod \ N \qquad (a)$$

Now, if α and β are the remainders of the division of integers A and B, respectively, by N, we can write:

$$A = IN + \alpha \longrightarrow \alpha = A \ Mod \ N$$
$$B = JN + \beta \longrightarrow \beta = B \ Mod \ N \qquad (b)$$

thus

$$AB = INJN + \alpha JN + \beta IN + \alpha\beta$$

but, if K is the quotient of the integer division of $\alpha\beta$ by N then

$$\alpha\beta = KN + \gamma \longrightarrow \gamma = (\alpha\beta) \ Mod \ N$$

and

$$AB = (INJ + \alpha J + \beta I + K) \ N + \gamma$$

thus

$$(AB) \ Mod \ N = \gamma = (\alpha\beta) \ Mod \ N \qquad (c)$$

or, according to (*b*)

$$(AB) \ Mod \ N = \left[(A \ Mod \ N) \ (B \ Mod \ N) \right] \ Mod \ N \qquad (d)$$

From (*b*)

$$A + B = (I + J) \ N + (\alpha + \beta)$$

but, if M and α are the quotient and the remainder, respectively of the division of $\alpha + \beta$ by N, it follows that

$$\alpha + \beta = MN + \delta \longrightarrow \delta = (\alpha + \beta) \ Mod \ N$$

thus

$$A + B = (I + J + M) N + \delta \quad \rightarrow \quad \delta = (A + B) \; Mod \; N$$

consequently

$$(A + B) \; Mod \; N = (\alpha + \beta) \; Mod \; N \qquad (e)$$

or, according to (*b*)

$$(A + B) \; Mod \; N = (A \; Mod \; N + B \; Mod \; N) \; Mod \; N \qquad (f)$$

Finally, if

$$\left. \begin{array}{c} P < N \\ P \; Mod \; N = P \end{array} \right. \qquad (g)$$

Expressions (*a*), (*d*), (*f*) and (*g*) are all we need to effect all the developments.

A P P E N D I X II

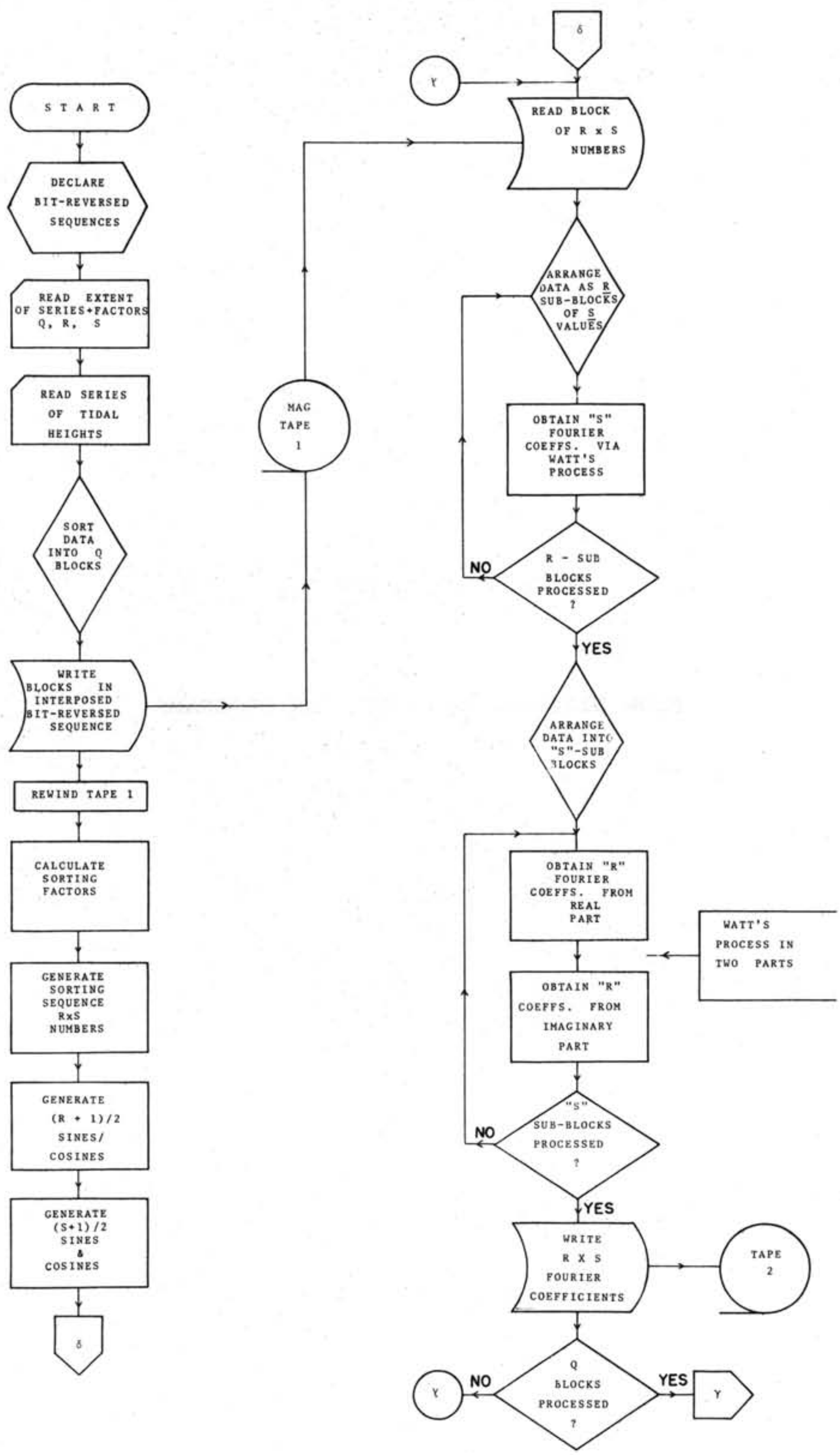
FLOW DIAGRAMS and COMPUTER PROGRAMS

FLOW DIAGRAM-I

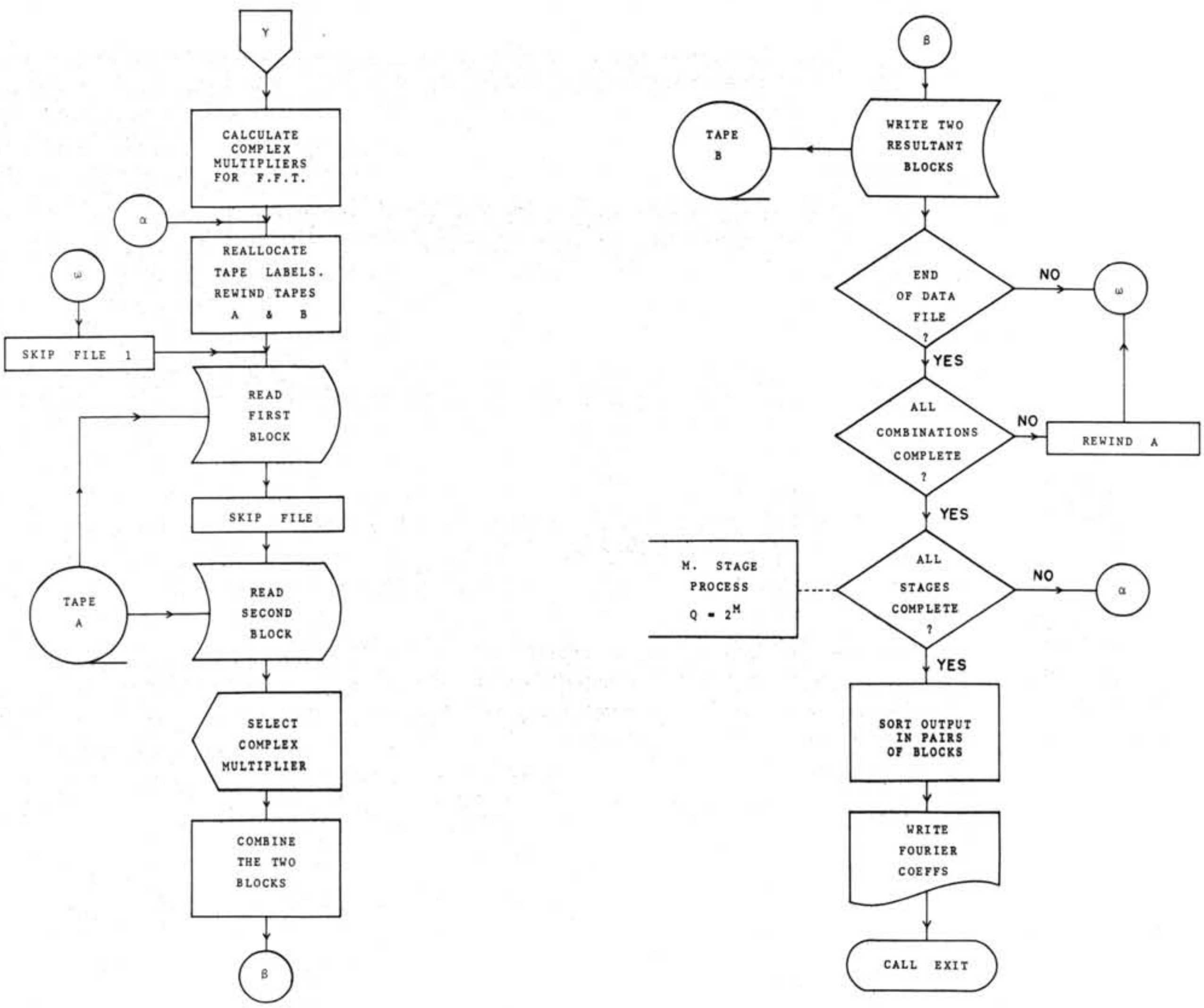
THOMAS PRIME FACTOR ALGORITHM

FOR THREE FACTORS UTILISING

THE F. F. T.

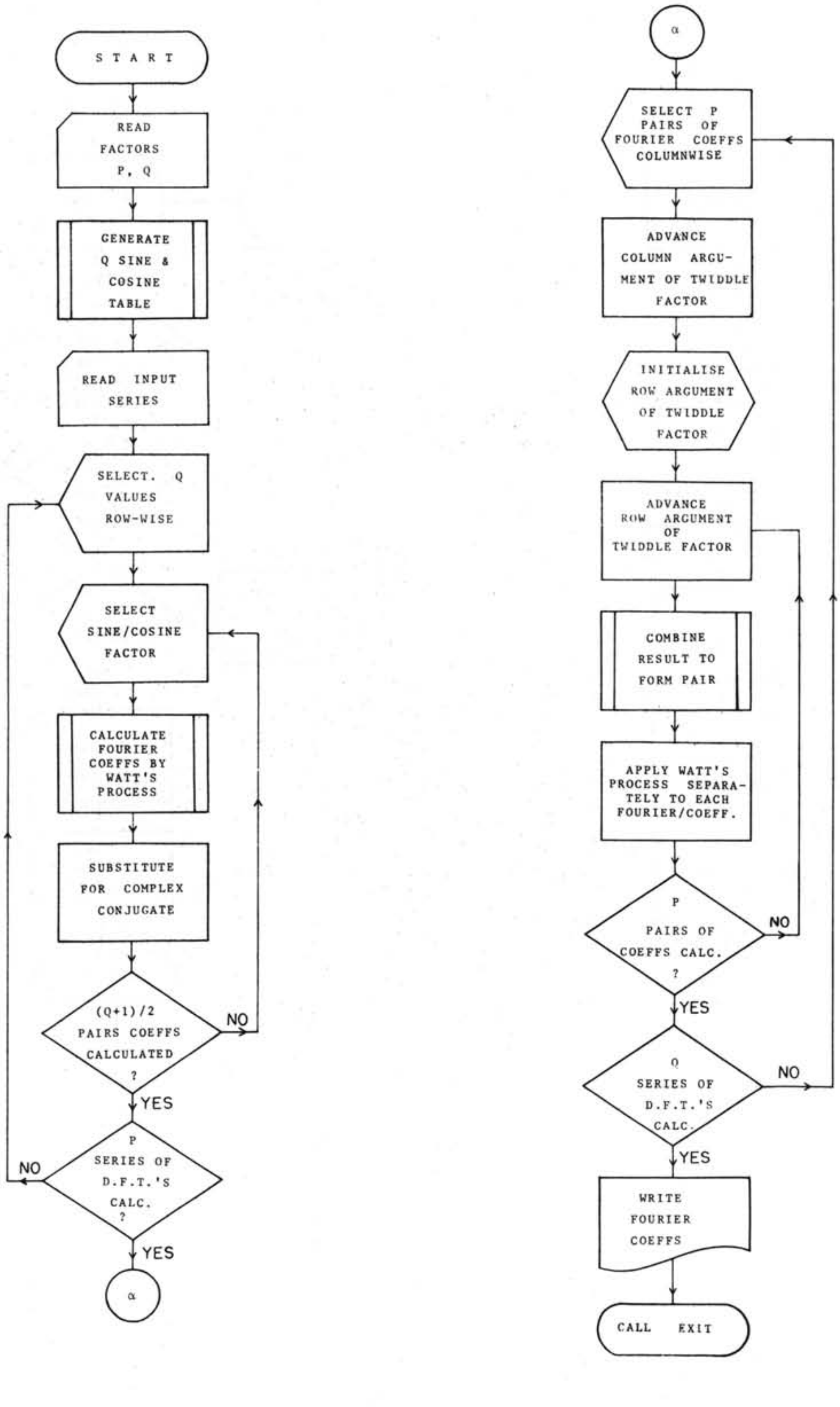


FLOW DIAGRAM-I
(CONTINUED)



FLOW DIAGRAM 2

GENTLEMAN & SANDE ALGORITHM



```
      IMPLICIT COMPLEX(V,W)
      INTEGER*2Y(8856),BETA(8),INV(4),GAMA
      INTEGER*2 KTAB(1107)
      DIMENSION SINQ(21),COSQ(21),SINP(14),COSP(14),Y(41),WBETA(4),
     1RA(41),RB(41)
      DIMENSION VV(4528)
      COMMON VA(1107),WA(1107),A(1108),B(1108),YY(1107),DUMMY(5733)
      EQUIVALENCE (VA(1),Y(1)),(VV(1),A(1))
      DATA BETA/0,1,4,5,2,3,6,7/
      DATA INV/0,2,1,3/
C     FAST FOURIER ANALYSIS OF TIDES ON MAGNETIC TAPE
C    USING THE THOMAS PRIME ALGORITHM IN CONJUNCTION WITH THE FFT
      READ(5,500)NFFT,NDAYS,IQ,IP
C    DATA= NO. OF FFT'S =2**M,  NO OF DAYS IN  SEQUENCE,FACTORS OF DFT'S
C   WHICH SHOULD BE ODD
C    ARRAY (INV) IS BIT-REVERSED SEQUENCE FOR 2**(M-1) NUMBERS
C    ARRAY (BETA) IS INTERPOSED BIT-REVERSED SEQUENCE FOR 2**M NUMBERS
  500 FORMAT(4I4)
      IPQ=IP*IQ
      LGRUP=0
      GAMA=3
      NT=IPQ*NFFT
      N=NDAYS*24
      IF(N.NE.NT)GOTO 299
      RN=2./N
      REWIND 2
      REWIND 3
      IE=0
C    DATA SERIES READ AS AN INTEGER ARRAY
      READ(5,501)(Y(I),I=1,N)
  501 FORMAT (24I3)
      ISUM=0
      DO 12 I=1,N
   12 ISUM=Y(I)&ISUM
      YSUM=ISUM
      YSUM=YSUM/N
      DO 14 II=1,NFFT
      KK=BETA(II)*IPQ-NFFT&1
C   STATEMENT TO HALF-BIT REVERSE SERIES
      DO 13 L=1,IPQ
      KK=KK&NFFT
      IF(KK.GT.N)KK=KK-N
   13 A(L)=Y(KK)-YSUM
   14 WRITE(2)(A(I),I=1,IPQ)
C   DATA STORED ON TAPE FOR SUCCESSIVE PASSES OF THOMAS PRIME ALGORITHM
      REWIND 2
      FACT=6.28318/IPQ
      IPQ1=IPQ-1
      IQ2=IQ&2
      IP2=IP&2
      IQ1=IQ-1
      IP1=IP-1
C    GENERATE SINE AND COSINE TABLES
      ARG=FACT*IP
      ANG=0.
      IHQ=(IQ&1)/2
      DO 10 J=1,IHQ
      SINQ(J)=SIN(ANG)
      COSQ(J)=COS(ANG)
   10 ANG=ANG&ARG
      ARG=FACT*IQ
```

```
      ANG=0.
      IHP=(IP&1)/2
      DO 20 J=1,IHP
      SINP(J)=SIN(ANG)
      COSP(J)=COS(ANG)
   20 ANG=ANG&ARG
C     CALCULATE TRIPLE SORTING FACTORS
      NP=NFFT*IP
      NQ=NFFT*IQ
      IMP=&1
    6 IMP=IMP&IP
      IF(MOD(IMP,NQ).NE.0)GOTO 6
      IMQ=&1
    7 IMQ=IMQ&IQ
      IF(MOD(IMQ,NP).NE.0)GOTO 7
      INFT=&1
    8 INFT=INFT&NFFT
      IF(MOD(INFT,IPQ).NE.0)GOTO 8
C     INFT IS DEFINED BY MOD(INFT,NFFT)=1 AND ALSO MOD(INFT,IP*IQ)=0  ETC.
      MM=-IMQ
      KK=0
C     CONSTRUCT SORTING TABLE FOR EACH BLOCK
      DO 24 J=1,IQ
      MM=MM&IMQ
      IF(MM.GE.N)MM=MM-N
      M=MM
      DO 24 I=1,IP
      KK=KK&1
      IF(M.GE.N)M=M-N
      KTAB(KK)=M
   24 M=M&IMP
C   APPLICATION OF THOMAS PRIME SUCCESSIVELY
   25 READ(2)(YY(I),I=1,IPQ)
      M=0
      DO 350 II=1,IPQ,IQ
      K=II-IP
      DO 110 L=1,IQ
      K=K&IP
      IF(K.GT.IPQ)K=K-IPQ
  110 X(L)=YY(K)*RN
C     DATA SORTED ON ENTRY FOLLOWING  Y(I,K)=X(IQ*M&IP*P)(M=0,IP-1$P=0,IQ- )
      JQ=M&IQ2
      DO 125 JJ=1,IHQ
      SINFI=SINQ(JJ)
      COSFI=COSQ(JJ)
      COSTH=COSFI&COSFI
C     THE NUMBER OF FACTORS TO BE EVALUATED MUST BE ODD
      U2=0.
      U1=X(IQ)
      I=IQ1
  120 U0=X(I)&U1*COSTH-U2
      U2=U1
      U1=U0
      I=I-1
      IF(I-1)121,121,120
  121 M=M&1
      A(M)=X(1)&COSFI*U1-U2
      B(M)=SINFI*U1
C     SUBSTITUTING THE COMPLEX CONJUGATES
      JQ=JQ-1
      A(JQ)=A(M)
```

```
  125 B(JQ)=-B(M)
C   RESETTING M
      M=M&IHQ-1
  350 CONTINUE
C     END OF FIRST BLOCK. NOW WATT'S FORMULA IS APPLIED TO COMPLEX COEFFICIENTS
      MM=1
      DO 450 II=1,IQ
      M=MM
      ML=M&IP
      L=0
      DO 360 K=II,IPQ,IQ
      L=L&1
      RA(L)=A(K)
  360 RB(L)=B(K)
      JJ=1
      DO 395 JJ=1,IHP
      SINFI=SINP(JJ)
      COSFI=COSP(JJ)
      COSTH=COSFI&COSFI
      U2=0.
      Q2=0.
      U1=RA(IP)
      Q1=RB(IP)
      I=IP1
  390 Q0=RB(I)&Q1*COSTH-Q2
      U0=RA(I)&COSTH*U1 -U2
      U2=U1
      Q2=Q1
      U1=U0
      Q1=Q0
      I=I-1
      IF(I-1)391,391,390
  391 AR=RA(1)&COSFI*U1-U2
      BR=RB(1)&COSFI*Q1-Q2
      AI=SINFI*U1
      BI=SINFI*Q1
C   COMBINING THE REAL AND IMAGINARY PARTS
      WA(M)=CMPLX(AR-BI,BR&AI)
      IF(JJ.NE.1)WA(ML)=CMPLX(AR&BI,BR-AI)
C   STATEMENT TO SORT THE COMPLEX CONJUGATE
      ML=ML-1
  395 M=M&1
  450 MM=MM&IP
      LGRUP=LGRUP&1
      WRITE(3)(WA(I),I=1,IPQ)
C  STORING THE FOURIER COEFFICIENTS FROM THE SUCCESSIVE PASSES ON TAPE
      IF(LGRUP.LT.NFFT) GOTO 25
      NFT4=NFFT/4
      NFT2=NFT4&NFT4
      NSTOP=N/2&1
      INF4=MOD(INFT*NFT2,N)
      MM=1
      FACT=6.28318/NFFT
C     GENERATION OF COMPLEX MULTIPLIERS FOR F.F.T.
      DO 151 I=1,NFT2
      ARG=FACT*INV(I)
  151 WBETA(I)=CMPLX(-COS(ARG),-SIN(ARG))
C   NOTE THE CHANGE OF SIGN IN THE COMPLEX MULTIPLIER TO FACILITATE THE
C   COMPUTATION
      ITAPE=3
      JTAPE=2
```

```
      NSTEP=NFT2
      LPASS=0
C  INTIALISE TAPE LABELS
  691 LPASS=LPASS&1
      REWIND ITAPE
      REWIND JTAPE
      LBLOC=0
      K=0
      GOTO 700
  695 READ(ITAPE)
  700 READ(ITAPE)(VA(I),I=1,IPQ)
      READ(ITAPE)
      READ(ITAPE)(WA(I),I=1,IPQ)
      LBLOC=LBLOC &1
      IF(LBLOC.EQ.NFT4)REWIND ITAPE
      K=K&1
      WBK=WBETA(K)
      IF(K.EQ.NSTEP)K=0
C   BLOCK FOR DETERMINING THE COMPLEX MULTIPLIER
C  COMBINING BLOCKS VIA F.F.T. ALGORITHM
  815 DO 820 I=1,IPQ
      V1=VA(I)
      VA(I)=WA(I)&V1
  820 WA(I)=(WA(I)-V1)*WBK
C   THE FORMULA IS CHANGED SLIGHTLY WITH THE SIGN STORED IN THE COMPLEX
C   MULTIPLIER
      IF(LPASS.EQ.GAMA)GOTO 720
      WRITE(JTAPE)(VA(I),I=1,IPQ)
      WRITE(JTAPE)(WA(I),I=1,IPQ)
      IF(LBLOC.NE.NFT2)GOTO 695
      JFT=ITAPE
      ITAPE=JTAPE
      JTAPE=JFT
      NSTEP=NSTEP/2
      GOTO 691
  720 KK=(LBLOC-1)*INFT&1
C  OUTPUT SORTED ACCORDING TO  K=INFT*MM & IMQ*II &IMP*JJ
C  MM=0,1,2....NFFT-1),(II=0,1,2,.....IQ-1),(JJ=0,1,2,.......IP-1)
      KK=MOD(KK,N)
      DO 830 I=1,IPQ
      II=KTAB(I)&KK
      IF(II.GT.N)II=II-N
      IF(II.LE.NSTOP)VV(II)=VA(I)
      JJ=II&INF4
      IF(JJ.GT.N)JJ=JJ-N
  830 IF(JJ.LE.NSTOP)VV(JJ)=WA(I)
      IF(LBLOC.NE.NFT2)GOTO 695
      REWIND 2
      WRITE(2)(VV(I),I=1,NSTOP)
      GOTO 301
  299 WRITE(6,600)
  600 FORMAT('  P & Q FACTORS ARE NOT CORRECT')
  301 CALL EXIT
      END
```

```
      IMPLICIT COMPLEX(W)
      INTEGER*2Y(695)
      DIMENSION WA(696),SINQ(70),COSQ(70),X(139),RA(5),XB(5)
C   METHOD-- ALGORITHM OF GENTLEMAN AND SANDE USING THE DIFFERENCE METHOD
C  OF WATT FOR REAL VALUED FOURIER SERIES
      READ(5,500)N,IP,IQ
C   DATA--N=NUMBER OF VALUES TO BE READ,OUTPUT IS OF N/2 &1 FOURIER COEFFICIENTS
C  IP,IQ ARE THE FACTORS OF N, WHICH CAN BE EVEN OR IDENTICAL
  500 FORMAT(3I4)
      IPQ=IP*IQ
      FACT=6.28318/IPQ
      IQ2=IQ&2
      IP2=IP&2
      IP1=IP-1
      IQ1=IQ-1
      IHP=(IP&1)/2
      IHQ=(IQ&1)/2
      IF(N.NE.IPQ)GOTO 299
      READ(5,501)(Y(I),I=1,N)
  501 FORMAT(24I3)
      ARG=FACT*IP
      ANG=0.
      DO 10 J=1,IHQ
      SINQ(J)=SIN(ANG)
      COSQ(J)=COS(ANG)
   10 ANG=ANG&ARG
C   END OF INITALISING THE SINE TABLES
      RN=2./N
      M=0
C  DATA SORTED ON ENTRY TO LOOP AND WATT'S PROCESS APPLIED
      DO 350 II=1,IP
      L=0
      DO 110 K=II,IPQ,IP
      L=L&1
  110 X(L)=Y(K)*RN
      JQ=M&IQ2
      DO 125 JJ=1,IHQ
      COSFI=COSQ(JJ)
      COSTH=COSFI&COSFI
      U2=0.
      U1=X(IQ)
      I=IQ1
  120 U0=X(I)&COSTH*U1-U2
      U2=U1
      U1=U0
      I=I-1
      IF(I.NE.1)GOTO 120
      M=M&1
      WA(M)=CMPLX(X(1)&COSFI*U1-U2,SINQ(JJ)*U1)
      JQ=JQ-1
  125 WA(JQ)=CONJG(WA(M))
  350 M=M&IHQ-1
C    END OF FIRST BLOCK. NOW WATT'S FORMULA IS APPLIED TO COMPLEX COEFFICIENTS
      MM=1
C   INTIALISING COLUMN ARGUEMENT OF TWIDDLE FACTOR
      ARGP=FACT*IQ
      ARG=0.
      DO 450 II=1,IQ
      M=MM
      L=0
      DO 360 K=II,IPQ,IQ
```

```
      L=L&1
      RA(L)=REAL(WA(K))
  360 XB(L)=AIMAG(WA(K))
C   INITIALISE ROW ARGUEMENT OF TWIDDLE FACTOR
      ANG=ARG
      DO 370 JJ=1,IP
      SINFI=SIN(ANG)
      COSFI=COS(ANG)
      COSTH=COSFI&COSFI
      U2=0.
      V2=0.
      U1=RA(IP)
      V1=XB(IP)
      I=IP1
  390 V0=XB(I)&COSTH*V1-V2
      U0=RA(I)&COSTH*U1-U2
      U2=U1
      V2=V1
      U1=U0
      V1=V0
      I=I-1
      IF(I-1)391,391,390
  391 WA(M)=CMPLX(RA(1)&COSFI*U1-SINFI*V1-U2,SINFI*U1&COSFI*V1&XB(1)-V2)
C   COMBINING REAL AND IMAGINARY PARTS
      WRITE(6,601)(I,WA(I),I=1,N)
      M=M&IQ
  370 ANG=ANG&ARGP
C  INCREMENT ROW ARGUEMENT
      MM=MM&1
  450 ARG=ARG&FACT
C  INCREMENT COLUMN ARGUEMENT
      WRITE(6,601)(I,A(I),B(I),I=1,N)
  601 FORMAT(4(1X,I4,1X,E12.4,E12.4))
  299 CALL EXIT
      END
```