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 continuous signals the transform pair may be written:

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

\[ g(t) = \int_{-\infty}^{\infty} G(f) e^{i2\pi ft} df \]  
(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. N° 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:

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

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

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 year's 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:

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

or using the notation:

\[ W_N = e^{-2\pi i/N} \quad (2b) \]

\[ c(n) = \sum_{k=0}^{N-1} X(k)w_N^nk \quad (n=0,1,2,\ldots,N-1) \quad (2a) \]

Now let k take the form \( k=(bp+j) \)

and let \( n=(a+m(N/p)) \)

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

\[ c(a+m(N/p)) = \sum_{j=0}^{p-1} \sum_{b=0}^{(N/p)-1} X(bp+j)W_{Np}^{(bp+j)(a+m(N/p))} \quad (2d) \]

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

Weing equation (2d) through, we get:

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

but, according to (2b),

\[ W_N = 1 \quad \text{and} \quad W_{Np} = e^{-2\pi i/(N/p)}W_{Np} \]

Thus rearranging

\[ c(a+m(N/p)) = \sum_{j=0}^{p-1} \sum_{b=0}^{(N/p)-1} X(bp+j)W_{Np}^{(bp+j)(a+m(N/p))}W_{Np}^{mbN} \quad (2e) \]

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

Let

\[ B_j(a) = \sum_{b=0}^{(N/p)-1} X(bp+j)w_{N/p}^b \]

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

\[ c(a+m(N/p)) = \sum_{j=0}^{p-1} B_a(j)w_{N}^{j(a+m(N/p))} \]

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

\[ w_{N}^{j(a+m(N/p))} = (w_{N}^{a}w_{N}^{m})^{j} \]

since according to (2b):

\[ w_{N}^{N/p} = w_{p} \]

from which it can be seen that the normal multiplier of the D.F.T. (viz: \( w_{N}^{m} \)) is multiplied by an additional corrective factor \( w_{N}^{a} \), called the "twiddle factor" by its origins; 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=2Y 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

\[ N = pq \]

(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:

\[ c(n) = \sum_{k=0}^{N-1} X(k)w_{N}^{nk} \]

(3b)

and put:

\[ k = (jp+mq) \mod N \]

\[ \begin{cases} j=0,1,2...q-1 \\ m=0,1,2...p-1 \end{cases} \]

(3c)

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

\[ 0 \leq k \leq N-1 \]

(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>
<thead>
<tr>
<th>( j )</th>
<th>( m )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td>12</td>
<td>15</td>
<td>18</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>7</td>
<td>10</td>
<td>13</td>
<td>16</td>
<td>19</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>14</td>
<td>17</td>
<td>20</td>
<td>2</td>
<td>5</td>
<td>8</td>
<td>11</td>
</tr>
</tbody>
</table>

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

\[
n = (gI+hJ) \mod N \quad \text{where} \quad g=0,1,2,\ldots p-1, h=0,1,2,\ldots q-1
\]

Before replacing \( k \) and \( n \) in \( w_{nk}^n \), by expressions (3c) and (3e) it is convenient to derive a general expression for \( W_L^M \), 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 \( \beta \) is the remainder of that division, we have according to (2b):

\[
w^L_M = e^{-i2\pi(KM+\beta)/M} = e^{-i2\pi K - i2\pi \beta/M} = \omega^\beta_M = w^L_M \mod M
\]

Consequently

\[
w_{nk}^n = w^n_M \mod N
\]

and from (3c) and (3e)

\[
w_{nk}^n = w^n_M [(jg+mq) \mod N (gI+hJ \mod N) \mod N]
\]

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

\[
w_{nk}^n = w^n_M (jgI+jhpI+mgqI+mhqI)
\]

But, from (2b) and (3a) we have:
\[ w_N^p = e^{-i2\pi p/pq} = w_q \]
and
\[ w_N^q = e^{-i2\pi q/pq} = w_p \]
thus
\[ w_{nk} = w_q^{ig I} w_q^{jh J} w_p^{mg I} w_p^{mh J} \]
(3g)

Since we can choose I and J at our convenience, these factors may be chosen to satisfy the following relationships:
\[ w_I^I = w_q Mod q = w_0 \]
\[ w_J^J = w_q Mod q = w_q \]
\[ w_I^p = w_p Mod p = w_p \]
\[ w_J^p = w_p Mod p = w_0 \]
(3h)

which means that I and J must be given by
\[ \begin{align*}
I Mod q &= 0 \\
I Mod p &= 1 \\
J Mod q &= 1 \\
J Mod p &= 0 \\
\end{align*} \]
(3i)

thus, according to (3h), expression (3g) reduces to:
\[ w_{nk} = w_q^{jh} w_p^{mg} \]
(3j)

Consequently, by using (3c), (3b), (3e) and (3j) we can change (3b) into:
\[ c \cdot (gI+hJ) Mod N = \sum_{j=0}^{q-1} \sum_{m=0}^{p-1} X \cdot (jp+mq) Mod N \cdot w_q^{jh} w_p^{mg} \]
or
\[ c \cdot (gI+hJ) Mod N = \sum_{j=0}^{q-1} \sum_{m=0}^{p-1} w_q^{jh} w_p^{mg} \cdot (jp+mq) Mod N \]

This expression can be split into two, by using a more suitable matrix notation:
\[ ||w_p^{mg}|| \{x_j(m)\} = \{a_j(g)\} \]
and
\[ ||w_q^{jh}|| \{a_g(j)\} = \{c_g(h)\} \]
Since we have $q$ values of $j$ and $p$ values of $g$, there will be $q$ groups of values of $c_j(g)$, 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 -** $n = (15q+7h) \mod 21$

<table>
<thead>
<tr>
<th>$h$</th>
<th>$0$</th>
<th>$1$</th>
<th>$2$</th>
<th>$3$</th>
<th>$4$</th>
<th>$5$</th>
<th>$6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0$</td>
<td>0</td>
<td>15</td>
<td>9</td>
<td>3</td>
<td>18</td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td>$1$</td>
<td>7</td>
<td>1</td>
<td>16</td>
<td>10</td>
<td>4</td>
<td>19</td>
<td>13</td>
</tr>
<tr>
<td>$2$</td>
<td>14</td>
<td>8</td>
<td>2</td>
<td>17</td>
<td>11</td>
<td>5</td>
<td>20</td>
</tr>
</tbody>
</table>

Another possibility exists to choose $I$ and $J$ so that

$$\begin{align*}
   I \mod q &= 0 \\
   I \mod p &= p-1 \\
   J \mod q &= q-1 \\
   J \mod p &= 0
\end{align*}$$

In this case it is easy to prove that the conjugates of $c(n)$ are found, i.e., $c(N-n)$, for $n=0,1,2, \ldots N-1$. 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

$$p = rs$$

we have

$$N = rsq$$

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

$$m = (ar+bs) \mod p$$

\[\begin{align*}
   a &= 0,1,2,\ldots s-1 \\
   b &= 0,1,2,\ldots r-1
\end{align*}\]
and

\[ n = (gI + hJ + lL) \mod N \]

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

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

\[ k = (jp + mq) \mod N = \{jp + [(ar + bs) \mod p]\} \mod N \]

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

\[ k = \{j(rs) + a(rq) + b(sq)\} \mod N \]

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 = \( r=3, q=4, s=5 \)

<table>
<thead>
<tr>
<th></th>
<th>j = 0</th>
<th>j = 1</th>
<th>j = 2</th>
<th>j = 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>20</td>
<td>40</td>
<td>15</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>12</td>
<td>32</td>
<td>52</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

**TABLE 4-II**

\[ n = (40g + 36h + 45l) \mod 60 \]

Output mapping for 3 factors = \( r=3, q=4, s=5 \)

<table>
<thead>
<tr>
<th></th>
<th>g = 0</th>
<th>g = 1</th>
<th>g = 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>h</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>45</td>
<td>30</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>36</td>
<td>21</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>12</td>
<td>57</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>48</td>
<td>33</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>24</td>
<td>9</td>
</tr>
</tbody>
</table>
Now from (4d) and (4e) we can obtain:

\[ w_{nk}^N = w_N^N \left[ j(rs) + a(rq) + b(sq) \right] g^{I+H+IL} \]

\[ = \left( w_{rs}^I \right)^j g \left( w_{rq}^J \right)^h \left( w_{sq}^L \right)^l \]

\[ \times \left( w_{rs}^I \right)^a g \left( w_{rq}^J \right)^b \left( w_{sq}^L \right)^c \]

\[ \times \left( w_{rs}^I \right)^b g \left( w_{rq}^J \right)^a \left( w_{sq}^L \right)^b \]

(4f)

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

\[ w_{rs}^N = e^{-i2\pi rs/rsq} w_q \]

\[ w_{rq}^N = e^{-i2\pi rq/rsq} w_s \]

\[ w_{sq}^N = e^{-i2\pi sq/rsq} w_r \]

thus

\[ w_{nk}^N = \left( w_{rs}^I \right)^j g \left( w_{rq}^J \right)^h \left( w_{sq}^L \right)^l \]

\[ \times \left( w_{rs}^I \right)^a g \left( w_{rq}^J \right)^b \left( w_{sq}^L \right)^c \]

\[ \times \left( w_{rs}^I \right)^b g \left( w_{rq}^J \right)^a \left( w_{sq}^L \right)^b \]

Since we can choose I, J and L so that

\[
\begin{align*}
I \mod q &= 0 \\
I \mod s &= 0 \\
I \mod r &= 1 \\
J \mod q &= 0 \\
J \mod r &= 0 \\
J \mod s &= 1 \\
L \mod s &= 0 \\
L \mod r &= 0 \\
L \mod q &= 1
\end{align*}
\]

(4g)

it follows that

\[ w_{nk}^N = w_{rs}^I w_{rq}^J w_{sq}^L w_{rs}^I w_{rq}^J w_{sq}^L \]

(4h)

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

\[ c \left( g^{I+H+IL} \mod N \right) = w_{rs}^I w_{rq}^J w_{sq}^L \sum_{a=0}^{q-1} \sum_{b=0}^{s-1} \sum_{r=0}^{r-1} \left[ j(r) + a(rq) + b(sq) \right] \mod N \]

or, by using a more suitable notation,

\[ c_{gh} \left( \ell \right) = \sum_{a=0}^{q-1} \sum_{b=0}^{s-1} \sum_{r=0}^{r-1} w_{rs}^I w_{rq}^J w_{sq}^L \]

\[ \times \left[ j_{\ell} \left( a \right) + \sum_{j=0}^{q-1} \sum_{a=0}^{s-1} \sum_{b=0}^{r-1} \left( b_{\ell} \left( a \right) \right) \right] w_{rs}^I w_{rq}^J w_{sq}^L \]

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

\[
\begin{align*}
\| w_{r}^{b} \| \{ x_{ja}^{(b)} \} &= \{ \alpha_{ja}^{(g)} \} \\
\| w_{s}^{ah} \| \{ \alpha_{jg}^{(a)} \} &= \{ \gamma_{jg}^{(h)} \} \\
\| w_{q}^{j} \| \{ \gamma_{gh}^{(j)} \} &= \{ c_{gh}(Z) \}
\end{align*}
\] (4\text{e})

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 (4\text{e}) as input data; whereas \( c_{gh}(Z) \) are the values of \( c(n) \) appearing in the output according to the order given by (4\text{d}). (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

<table>
<thead>
<tr>
<th>Span in days</th>
<th>Span in hours</th>
<th>Factors</th>
<th>Number of cycles per series</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>( M_{2} )</td>
</tr>
<tr>
<td>15.0</td>
<td>360</td>
<td>5x9x8</td>
<td>28.984</td>
</tr>
<tr>
<td>29.0</td>
<td>696</td>
<td>29x3x8</td>
<td>56.036</td>
</tr>
<tr>
<td>58.0</td>
<td>1392</td>
<td>29x3x16</td>
<td>112.072</td>
</tr>
<tr>
<td>87.0</td>
<td>2088</td>
<td>29x9x8</td>
<td>165.108</td>
</tr>
<tr>
<td>104.5</td>
<td>2508</td>
<td>33x19x4</td>
<td>201.923</td>
</tr>
<tr>
<td>133.5</td>
<td>3204</td>
<td>29x27x4</td>
<td>257.959</td>
</tr>
<tr>
<td>162.5</td>
<td>3900</td>
<td>39x5x4</td>
<td>313.994</td>
</tr>
<tr>
<td>177.5</td>
<td>4260</td>
<td>71x15x4</td>
<td>342.979</td>
</tr>
<tr>
<td>192.5</td>
<td>4620</td>
<td>35x33x4</td>
<td>371.963</td>
</tr>
<tr>
<td>220.5</td>
<td>5292</td>
<td>49x27x4</td>
<td>460.066</td>
</tr>
<tr>
<td>235.5</td>
<td>5652</td>
<td>157x3x4</td>
<td>455.050</td>
</tr>
<tr>
<td>279.5</td>
<td>6708</td>
<td>43x39x4</td>
<td>540.070</td>
</tr>
<tr>
<td>297.0</td>
<td>7128</td>
<td>31x11x8</td>
<td>537.885</td>
</tr>
<tr>
<td>325.0</td>
<td>7800</td>
<td>39x25x8</td>
<td>627.929</td>
</tr>
<tr>
<td>355.0</td>
<td>8520</td>
<td>71x15x8</td>
<td>685.957</td>
</tr>
<tr>
<td>369.0</td>
<td>8856</td>
<td>41x27x8</td>
<td>713.009</td>
</tr>
</tbody>
</table>

Table 5-1 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 Wart (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

$$Y(k) = Y(k) + 2 \cos \theta \cdot X_{k+1} - X_{k+2} \quad (6\alpha)$$

$Y(k)$ 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

$$a_n = \frac{(X_0 - X_1 \cos \theta)}{2/N}$$

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

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 (2f) for each of the $N/p$ series $B_a(j) = (a=0,1,2,\ldots (N/p-1))$ the argument $\alpha$ of the "twiddle factor" $W^a_j$ is equivalent to a phase-shifting of the complex multiplier $W^m_j = W^a_j N/p$ which necessitates a recalculation of the multiplier $W^a_j n(N/p)$ 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 $W^a_j$ can be calculated at the start of the program, stored, and multiplied directly with the coefficients $B_a(j)$. 


89
FIG. 1 — Signal flow diagram for the F.F.T. treatment of $q=2^3$ blocks of D.F.T. 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 (4i) 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>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^1$</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$2^2$</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$2^3$</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>6</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$2^4$</td>
<td>0</td>
<td>8</td>
<td>4</td>
<td>12</td>
<td>2</td>
<td>10</td>
<td>6</td>
<td>14</td>
<td>1</td>
<td>9</td>
<td>55</td>
<td>13</td>
<td>3</td>
<td>11</td>
<td>7</td>
</tr>
</tbody>
</table>

NOTE: Each successive 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 roles 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 output, 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 circumstances 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 algoritmo 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 algoritmo de fatores primos de Thomas e o de Gentleman e Sande são, respectivamente, tratados em detalhe, na transformação de séries com número ímpar 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**


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

From the definition itself of $A \mod N$, it follows that

$$(((A \mod N) \mod N) \mod N \ldots)) = A \mod N$$  \hspace{1cm} (a)

Now, if $\alpha$ and $\beta$ are the remainders of the division of integers $A$ and $B$, respectively, by $N$, we can write:

$$A = IN + \alpha \quad \rightarrow \quad \alpha = A \mod N$$  \hspace{1cm} (b)

$$B = JN + \beta \quad \rightarrow \quad \beta = B \mod N$$

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

$$\alpha \beta = KN + \gamma \quad \rightarrow \quad \gamma = (\alpha \beta) \mod N$$

and

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

thus

$$(AB) \mod N = \gamma = (\alpha \beta) \mod N$$  \hspace{1cm} (c)

or, according to (b)

$$(AB) \mod N = \left[ (A \mod N) (B \mod N) \right] \mod N$$  \hspace{1cm} (d)

From (b)

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

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

$$\alpha + \beta = MN + \delta \quad \rightarrow \quad \delta = (\alpha + \beta) \mod N$$
thus

$$A + B = (I + J + M) N + \delta \quad \Rightarrow \quad \delta = (A + B) \mod N$$

consequently

$$(A + B) \mod N = (\alpha + \beta) \mod N \quad (e)$$

or, according to (b)

$$(A + B) \mod N = (A \mod N + B \mod N) \mod N \quad (f)$$

Finally, if

$$P < N$$

$$P \mod N = P \quad (g)$$

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

FLOW DIAGRAMS and COMPUTER PROGRAMS
FLOW DIAGRAM-I

THOMAS PRIME FACTOR ALGORITHM
FOR THREE FACTORS UTILISING
THE F.F.T.

START

DECLARE BIT-REVERSED SEQUENCES

READ EXTENT OF SERIES-FACTORS G, R, S

READ SERIES OF TIDAL HEIGHTS

SORT DATA AND INSERT BLOCKS

WRITE BLOCKS IN INTERPOSED BIT-REVERSED SEQUENCE

REVISE TAPE 1

CALCULATE SORTING FACTORS

GENERATE SORTING SEQUENCE #35 NUMBERS

GENERATE \((8 + 1)/2\) SINES/COSINES

GENERATE \((2+1)/2\) SINES/COSINES

READ BLOCK OF \(R \times S\) NUMBERS

ARRANGE DATA AS \(G\) SUB-BLOCKS OF \(S\) VALUES

OBTAIN "5" FOURIER COEFFS., VIA WATT'S PROCESS

YES

ARRANGE DATA INTO "5"-SUB-BLOCKS

NO

\(R - S\) BLOCKS PROCESSED?

NO

\(R\) COEFFS. FROM REAL PART

\(S\) COEFFS. FROM IMAGINARY PART

\(R-S\) BLOCKS PROCESSED?

YES

WRITE \(R \times S\) FOURIER COEFFICIENTS

\(R\) BLOCKS PROCESSED?

YES

TAKE 2

NO

CALCULATE COMPLEX MULTIPLIERS FOR F.F.T.

REALLOCATE TAPE LABELS, REVIVE TAPE A & B

SKIP FILE

TAPE A

READ FIRST BLOCK

SELECT COMPLEX MULTIPLIER

COMBINE THE TWO BLOCKS

B

TAPE B

WRITE TWO RESULTANT BLOCKS

END OF DATA FILE

ALL COMBINATIONS COMPLETE ?

NO

YES

REVERT A

NO

YES

N STAGE PROCESS $Q = 2^N$

ALL STAGES COMPLETE ?

NO

SORT OUTPUT IN PAIRS OF BLOCKS

WRITE FOURIER COEFFS

CALL EXIT

FLOW DIAGRAM -1
(CONTINUED)
FLOW DIAGRAM 2
GENTLEMAN & SANDE ALGORITHM

START
READ FACTORS P, Q
GENERATE Q SINE & COSINE TABLE
READ INPUT SERIES
SELECT Q VALUES ROW-WISE
SELECT SINE/COSINE FACTOR
CALCULATE FOURIER COEFFS BY WATT'S PROCESS
SUBSTITUTE FOR COMPLEX CONJUGATE

(Q+1)/2 PAIRS COEFFS CALCULATED?

P SERIES OF D.P.T.'S CALC.?

YES
NO

NO

YES

CALL EXIT

SELECT P PAIRS OF FOURIER COEFFS COLUMNWISE
ADVANCE COLUMN ARGUMENT OF TWIDDLE FACTOR
INITIALISE ROW ARGUMENT OF TWIDDLE FACTOR
ADVANCE ROW ARGUMENT OF TWIDDLE FACTOR
COMBINE RESULT TO FORM PAIR
APPLY WATT'S PROCESS SEPARATELY TO EACH FOURIER/COEFF.

P PAIRS OF COEFFS CALC.?

YES
NO

YES

SERIES OF D.P.T.'S CALC.?

WRITE FOURIER COEFFS
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),
IRA(41),R8(41)
DIMENSION VV(4528)
COMMON VA(1107),WA(1107),A(1108),3(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(414)
   IPQ=IP*IQ
   LGRP=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(2413)
   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   ALL=Y(KK)-YSUM
14   WRITE(2)(ALL,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=IQ62
   IP2=IP&2
   IO1=IQ-1
   IF=IP-1
C GENERATE SINE AND COSINE TABLES
ARG=FACT*IP
ANG=0.
   IHQ=(IOQ1)/2
   DO 10 J=1,IHQ
   SINQ(J)=SIN(ARG)
   COSQ(J)=COS(ARG)
10   ARG=ANG&ARG
   ARG=FACT*IQ
ANG=0.
IHP=IP$1$/2
DO 20 J=1,IHP
SIN(J)=SIN(ANG)
COSP(J)=COS(ANG)
20 ANG=ANG+ARG
C CALCULATE TRIPLE SORTING FACTORS
NP=NF$T*IP$
NQ=NF$T*IQ$
IMP=1$
6 IMP=IMP&IP
IF(MOD(IMP,NQ).NE.0)GOTO 6
IMQ=IMQ&IQ
IF(MOD(IMQ,NP).NE.0)GOTO 7
INFT=1$
8 INFT=INFT&NF$T$
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=-1IMQ
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&I
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 XL=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
SINF=SINO(JJ)
COSF=COSQ(JJ)
COSTH=COSF*GOSFI
C THE NUMBER OF FACTORS TO BE EVALUATED MUST BE ODD
U2=0.
JJ=IQ1
120 U0=X(IQ)
U1=U0
I=IQ1
121 M=M&1
AM=X(I)&GOSFI*U1-U2
BM=SINFI*U1
C SUBSTITUTING THE COMPLEX CONJUGATES
JQ=JQ-1
A(JQ)=AM

125 B(JQ)=-B(M)
C  END OF FIRST BLOCK. NOW WATT'S FORMULA IS APPLIED TO COMPLEX COEFFICIENTS
M=M&1
350 DO 450 II=1,IQ
     M=M&1
     ML=M&1P
     L=0
     DO 360 K=II,IPQ,IQ
     L=L&1
     RA(L)=A(K)
     RB(L)=B(K)
360 JJ=1
     DO 395 JJ=1,IPQ
     SINF=OSINP(JJ)
     COSF=OCOSP(JJ)
     COSTH=COSH*OSINF
     U2=0.
     Q2=0.
     U1=RA(JP)
     Q1=RB(JP)
     I=IP1
     Q0=RB(I)Q1*OSINH-Q2
     U0=RA(I)&OSINH*U1-U2
     U2=U1
     Q2=Q1
     U1=U0
     Q1=Q0
     I=I-1
     IF(I-1).LT.391,391,390
390 AR=RA(I)QOSINF&U1-U2
     BR=RB(I)QOSINF&Q1-Q2
     AI=OSINF&U1
     BI=OSINF&Q1
C  COMBINING THE REAL AND IMAGINARY PARTS
     WA(M)=CMPLX(AR-BI,BR-AI)
     IF(JJ.NE.1)WA(M)=CMPLX(AR&BI,BR-AI)
C  STATEMENT TO SORT THE COMPLEX CONJUGATE
     ML=ML&1
395 M=M&1
450 DO 450 II=1,IQ
     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(NFFT*NFT2,N)
M' 1
FACT=6.28318/NFFT
C  GENERATION OF COMPLEX MULTIPLIERS FOR F.F.T.
     DO 151 I=1,NFFT
     ARG=FACT&INV(I)
     WBIET(I)=CMPLX(-OSIN(ARG),-OSIN(ARG))
C  NOTE THE CHANGE OF SIGN IN THE COMPLEX MULTIPLIER TO FACILITATE THE
C  COMPUTATION
     ITAPE=3
     JTAPE=2
NSTEP=NFT2
L Pass=O
C INITIALISE TAPE LABELS
   691 L Pass=L Pass&1
   REWIND ITAPE
   REWIND JTAPE
   LBLOC=O
   K=O
   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=O
C BLOCK FOR DETERMINING THE COMPLEX MULTIPLIER
C COMBINING BLOCKS VIA F.F.T. ALGORITHM
815 DO 820 I=1,IPQ
     VI=VA(I)
820 WA(I)=(WA(I)-VI)*WBK
C THE FORMULA IS CHANGED SLIGHTLY WITH THE SIGN STORED IN THE COMPLEX
C MULTIPLIER
   IF(L Pass.EQ.GAMA)GOTO 720
   WRITE(JTAPE)(VA(I),I=1,IPQ)
   WRITE(JTAPE)(WA(I),I=1,IPQ)
   IF(LBLOC.EQ.NFT2)GOTO 695
   JFT=ITAPE
   ITAPE=JTAPE
   JTAPE=JFT
   NSTEP=NSTEP&1
   GOTO 691
720 KK=(LBLOC-1)*INFT&1
C OUTPUT SORTED ACCORDING TO K=INFT*M4 & IMQ*II & IMP*JJ
   MM=O,1,2,...,NFFT-1,(II=O,1,2,...,I-1),(JJ=O,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&INFT4
   IF(JJ.GT.N)JJ=JJ-N
830 IF(JJ.LE.NSTOP)VV(JJ)=WA(I)
   IF(LBLOC.EQ.NFT2)GOTO 695
   REWIND 2
   WRITE(2)(VV(I),I=1,NSTOP)
   GOTO 301
299 WRITE(6,600)
600 FORMAT(1P & Q FACTORS ARE NOT CORRECT*)
301 CALL EXIT
END
IMPLICIT COMPLEX(W)
INTEGER*2,VI695'
DIMENSION WA(696'),SINQ(70),COSQ(70),X(139),RA(5),X8(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(314)
   IPQ=IP*IQ
   FACT=6.28318/IPQ
   IQ2=IQ*IQ
   IP2=IP*IP
   IP1=IP-1
   IQ1=IQ-1
   IHP={IP*IQ}/2
   IQ2={IQ*IQ}/2
   IF(N.NE.IPQ)GOTO 299
READ(5,501)(Y(I),I=1,N)
501 FORMAT(24I3)
   ARG=FACT*IP
   ANG=O.
   DO 10 J=1,IHQ
      SIN(J)=SIN(ANG)
      COS(J)=COS(ANG)
      ANG=ANG+ARG
   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=O
   DO 110 K=II,IPQ,IP
      L=L+1
   110 X(L)=Y(K)*RN
   JQ=M&IQ2
   DO 125 JJ=1,IPQ
      COSFI=COSQ(JJ)
      COSTH=COSFI&COSFI
      U2=O.
      UI=X(IQ)
      I=I+1
   120 UO=X(I)&COSTH*UI-U2
      U2=UI
      UI=UO
      I=I-1
   IF(I.NE.1)GOTO 120
   M=M&1
   WA(M)=CMPLX(X(1)&COSFI*U1-U2,SINQ(JJ)*UI)
   JQ=JQ-1
350 M=M&IQ2
   IF(JQ.NE.1)GOTO 125
   WA(JQ)=CONJG(WA(M))
   M=MM
C END OF FIRST BLOCK. NOW WATT'S FORMULA IS APPLIED TO COMPLEX COEFFICIENTS
MM=1
C INITAILISING COLUMN ARGUEMENT OF TWIDDLE FACTOR
ARGP=FACT*I
ARG=O.
DO 450 II=1,I
   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 ARGUMENT OF TWIDDLE FACTOR
   ANG=ARG
   DO 370 JJ=1,IP
   SINFI=SIN(ANG)
   COSFI=COS(ANG)
   COSTH=COSFI&COSFI
   U2=0.
   V2=0.
   UI=RA(IP)
   V1=XB(IP)
   I=IP1
   390 VO=XB(I)&COSTH*V1-V2
   UO=RA(I)&COSFI*U1-U2
   U2=UL
   V2=V1
   UI=U0
   V1=V0
   I=I-1
   IF(I-1)391,391,390
   391 WA(M)=CMPLX(RA(I)&COSFI*UL-SINFI*U1-U2,SINFI*UL&COSFI*V1&XB(I)-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 ARGUMENT
   MM=MM&1
   450 ARG=ARG&FACT
   C INCREMENT COLUMN ARGUMENT
   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