AD-A160 546   FAST FOURIER TRANSFORM (FFT) SUBROUTINE FOR DETERMINING   1/1
FREQUENCY RESPONS..(U) ARMY MISSILE COMMAND REDSTONE
ARSENAL AL GUIDANCE AND CONTROL.. L BEACH JAN 85
UNCLASSIFIED   AMSMI/RG-85-15-TR SBI-AD-E950 763   F/G 12/1   NL

END
FILMED
DTIC

1.0 2.8 2.5 4.5 5.0 3.2 2.2 5.6 6.3 3.6 7.1 8.0 4.0 2.0 9.0 10 11.2 1.1 1.8 1.25 1.4 1.6

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

TECHNICAL REPORT-RG-85-15

AD-A160 546

FAST FOURIER TRANSFORM (FFT) SUBROUTINE FOR DETERMINING FREQUENCY RESPONSE DATA FOR DIGITAL SIMULATIONS

Linda Beach
Guidance and Control Directorate
Research, Development, and Engineering Center

January 1985

U.S. ARMY MISSILE COMMAND
Redstone Arsenal, Alabama 35898-5000

Approved for public release; distribution unlimited.

DISPOSITION INSTRUCTIONS

DISCLAIMER

TRADE NAMES

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TR-RG-85-15	2. GOVT ACCESSION NO. AD-A160546	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) FAST FOURIER TRANSFORM (FFT) SUBROUTINE FOR DETERMINING FREQUENCY RESPONSE DATA FOR DIGITAL SIMULATIONS		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Linda Beach		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Commander, US Army Missile Command ATTN: AMSMI-RG Redstone Arsenal, AL 35898-5254		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Same as above.		12. REPORT DATE January 1985
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

RADIX-2 algorithms
Fast Fourier Transform (FFT)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A FFT computer subroutine using VAX-11 FORTRAN has been written to perform the FFT. The FFT algorithm used to write the FFT subroutine is an in-place, decimation in frequency, Radix-2 algorithm originally proposed by Gentlemen and Sande. The subroutine can be linked with a system simulation to provide the frequency spectrum impulse data as a part of the system simulation. The FFT subroutine is a very useful, fast computational algorithm which can be used with any digital system simulation when frequency spectrum processing is needed in the calculation of the system's frequency response. This report outlines

(Continued)

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (Continued)

the development and checkout of the FFT and computer subroutine.

TABLE OF CONTENTS

A1

SYMBOLS

N	Number of input samples
M	Number of partial transformations
f	Continuous - frequency variable
f_h	Highest frequency in a spectrum
f_s	Sampling rate or frequency
F	Frequency increment between successive components
t	Continuous time variable
t_p	Effective period for a time function when it is periodic
T	Time increment between successive samples
$X_o(\ell)$	The set of input data samples
$X_m(\ell)$	- Where ℓ=0, 1, 2, ..., N-1 and m=1, 2, ...M is an output sequence of the mth nodal column computations
$X_m(\ell)$	- Becomes the input array during the $(m+1)^{st}$ stage of computations where $X_{m+1}(\ell)$ is the output array
ω_0	Fundamental frequency
K	Number of samples per cycle of f(t)

I. INTRODUCTION

In recent years the use of digital simulation has become an intricate part of military and civilian projects. A significant number of these projects involve digital system simulations which require frequency response analysis of the ensuing discrete-time, discrete-frequency simulation output. Very often these outputs are not composed of simple frequency sinewaves, but rather contain extraneous harmonics along with the desired fundamental frequency, ω_0, making it impractical to calculate amplitude ratios over the desired frequency range. When this is the case, the frequency response of the system can be obtained from the input-output amplitude ratios of the input and output frequency spectrums at each fundamental frequency, ω_0, over the desired frequency range.

An ideal method of generating the frequency spectrum of a discrete-time, discrete-frequency signal of a computer simulation is a Fast Fourier Transform (FFT) on the simulation output data. The FFT generates a discrete frequency spectrum analogous to the Fourier spectrum. Both give two impulses at $+\omega_0$ and $-\omega_0$ for a sinewave input. The frequency response of a discrete system simulation can be found by inputting a sinewave into the discrete system simulation and using the frequency spectrum input and output impulse ratios at ω_0 over the desired frequency range.

A FFT computer subroutine using VAX-11 FORTRAN has been written to perform the FFT. The subroutine can be linked with a system simulation to provide the frequency spectrum impulse data as a part of the system simulation. This report outlines the development and checkout of the FFT routine.

II. BACKGROUND

The FFT is an algorithm for computing the discrete Fourier transform of discrete data samples. There are many available FFT algorithms. The Radix-2 FFT is the one most commonly used. It is based on representing an array of size $N=2^M$ as a product of M factors, each of which is equal to 2. Radix-2 FFT algorithms are derived by decomposing the discrete Fourier transform into successively smaller discrete Fourier transforms. The manner of the decomposition produces the variation found in Radix-2 algorithms. Most of these algorithms may be classified as follows:

A. Decimation in Frequency

   1. In-place algorithm or

   2. Natural input-output algorithm

B. Decimation in Time

   1. In-place algorithm or

   2. Natural input-output algorithm

The FFT algorithm used to write the FFT subroutine is an in-place decimation in frequency, Radix-2 algorithm originally proposed by Gentlemen and Sande.

The algorithm is implemented in the following steps:

a. Initialization

b. A sequence of transformations, one partial transformation for each factor.

c. An unscrambling procedure.

III. DERIVATION OF A RADIX-2 FFT BY DECIMATION IN FREQUENCY

The discrete Fourier transform of $\{x(n)\}$ is a periodic sequence of complex numbers $\{X(k), k = 0,1,..., N-1\}$, defined by

$$X(k) = \sum_{n=0}^{N-1} x(n)\, W^{nk} \tag{1}$$

where

$$W = \exp\left(-j\frac{2\pi}{N}\right) = \cos\left(\frac{2\pi}{N}\right) - j \sin\left(\frac{2\pi}{N}\right)$$

Dividing the input sequence into two halves gives

$$X(k) = \sum_{n=o}^{\left(\frac{N}{2}\right)-1} x(n)\, W^{nk} + \sum_{n=N/2}^{N-1} x(n)\, W^{nk} \tag{2}$$

$$X(k) = \sum_{n=o}^{\left(\frac{N}{2}\right)-1} x(n)\, W^{nk} + W^{\left(\frac{N}{2}\right)k} \sum_{n=o}^{\left(\frac{N}{2}\right)-1} x\left(n + \frac{N}{2}\right) W^{-nk} \tag{3}$$

Combining the two summations in equation 3 and using the fact that $W^{\left(\frac{N}{2}\right)k} = (-1)^k$, yields

$$X(k) = \sum_{n=o}^{(\frac{N}{2})-1} \left(x(n) + (-1)^k \; x\left(n + \frac{N}{2}\right)\right) W^{nk} \quad (4)$$

Since $(-1)^k$ is equal to 1 for even K and equal to -1 for odd K, let even K = 2r and odd K = 2r = 1.

Dividing the sequence in equation 4 into an even and odd sequence yields a decimation in frequency of

$$X(2r) = \sum_{n=o}^{(\frac{N}{2})-1} \left(x(n) + x\left(n + \frac{N}{2}\right)\right) W^{2rn}$$

$$X(2r+1) = \sum_{n=o}^{(\frac{N}{2})-1} \left(x(n) - x\left(n + \frac{N}{2}\right)\right) W^{n} \; W^{2rn} \quad (5)$$

The arrays X(2r) and X(2r + 1) are the $\frac{N}{2}$ points DFT of the input arrays shown in equation (5). The signal flow graph for an eight-point input (N=8) is shown in Figure 1.

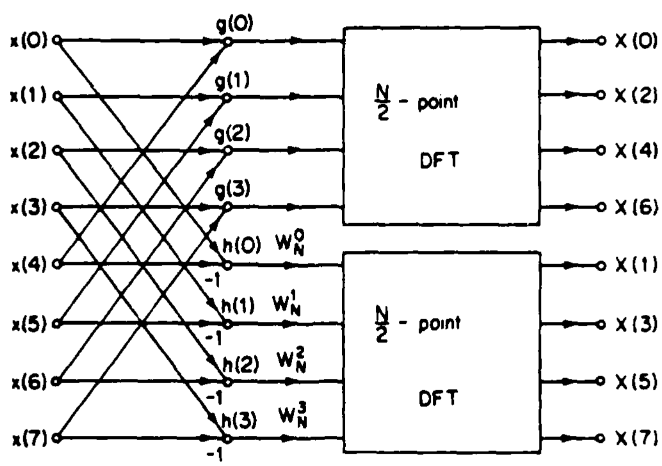

Figure 1. Flow graph of the decimation-in-frequency decomposition of an N-point DFT computation into two N/2-point DFT computations N=8)

The bit-reversed output can also be determined by the following method:

A. Write out the sequence of integer index numbers.

B. Convert the sequence to binary form.

C. Reverse the order of all the bits in each of the binary index numbers.

D. Convert back to integer index numbers.

Table 1 shows the determination of the bit-reversed output for the N=8 example using the method just described.

TABLE 1. Index Integers and Their Bit-Reversed Output Integers For N=8

Index Integer	0	1	2	3	4	5	6	7
Binary Index	000	001	010	011	100	101	110	111
Binary Bit-Reversed	000	100	010	110	001	101	011	111
Bit-Reversed Index Integer	0	4	2	6	1	5	3	7

Repetition of this decomposition for N=8 leads to Figures 2 and 3.

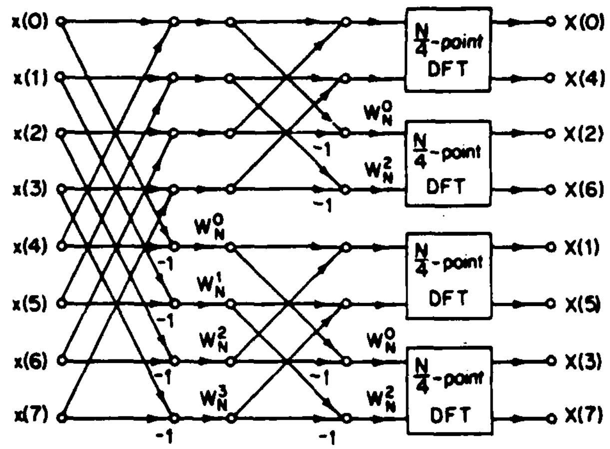

Figure 2. <u>Flow graph of decimation-in-frequency decomposition of an eight-point DFT into four two-point DFT computations.</u>

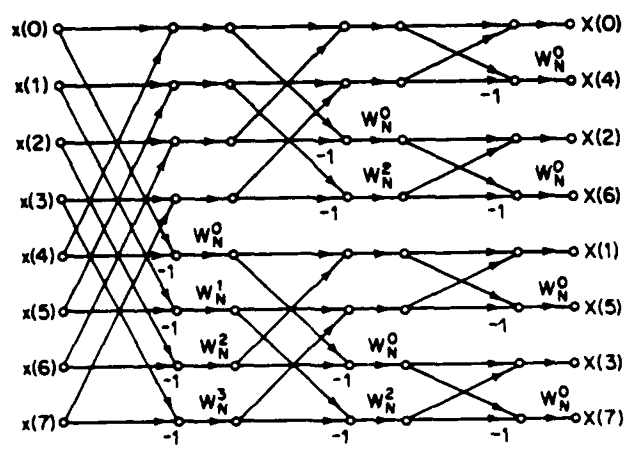

Figure 3. <u>Flow graph of complete decimation-in-frequency decomposition of an eight-point DFT computation.</u>

The preceding Figures show that the input sequence is in a natural order and the output sequence is in an unnatural order. This input-output order is referred to as natural input, bit-reversed output. It is a result of decomposition. Recall that originally the input sequence is divided into even-numbered samples and odd-numbered samples with the even-numbered samples in the first half and the odd-numbered samples in the second half. Such separations are carried out by examining the lease significant bit of the binary index representation where a zero corresponds to an even number and a one corresponds to an odd number. When the even and odd subsequences are sorted into their even and odd parts, the second least significant bit of the binary index representation is examined. This process is repeated until N subsequences of length 1 are obtained. This sorting process for N = 8 is shown in Figure 4.

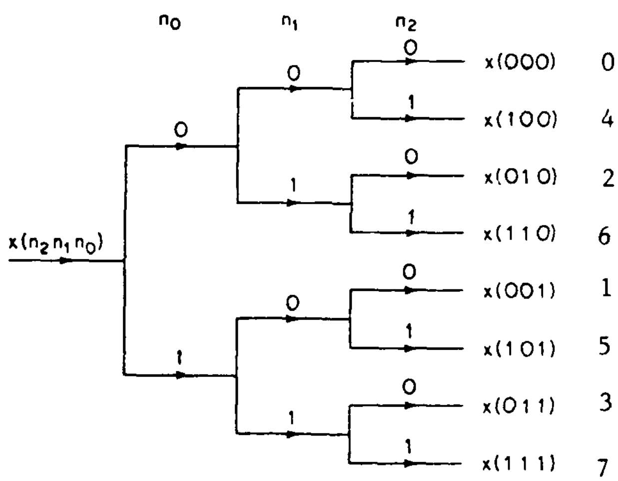

Figure 4. Bit-reversed sorting for N=8.

IV. COMPUTATIONS

The vertical nodes in the preceding flow graphs correspond to successive inner column computations. Each inner column computation takes a set of N complex numbers and transforms them into another set of N complex numbers. This process is repeated $M = \log_2 N$ times.

The flow graph for the basic computation (the butterfly computation) is shown in Figure 5.

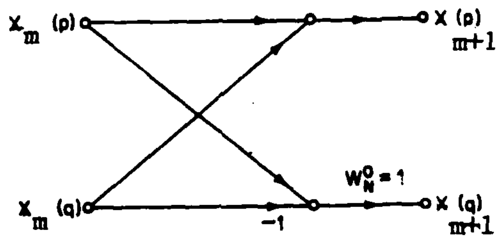

Figure 5. Flow graph of a typical two-point DFT as required in the last stage of decimation-in-frequency decomposition.

The set of equations found from the flow graph shown in Figure 5 is as follows

$$X_{m+1}(p) = X_m(p) + X_m(q)$$
$$X_{m+1}(q) = X_m(p) - X_m(q) \quad W_n^r \tag{6}$$

where

$$m = 0, 1, \ldots M - 1$$
$$p = 1, 2, \ldots N/2$$
$$q = \frac{N}{2} + 1, \ldots N$$

These equations are the algorithm for the transformations necessary for the Radix-2 FFT. Looking at equation (6), the data pair ($X_m(p)$, $X_m(q)$) is used once to compute the new data pair ($X_{m+1}(p)$, $X_{m+1}(q)$) and is not used again. Thus the transformations defined by equation (6) can be performed in place where each new transformation is stored in the same location occupied by the preceding transformation. This can be seen graphically in Figure 3 by the horizontal lines which connect consecutive nodes.

As an example of the use of the set of transformation equations, the first inner column computations of the 8-point FFT as shown in Figure 5 are found to be:

$$X_1(0) = X_o(0) + X_o(4)$$

$$X_1(1) = X_o(1) + X_o(5)$$

$$X_1(2) = X_o(2) + X_o(6)$$

$$X_1(3) = X_o(3) + X_o(7)$$

$$X_1(4) = W_N^o \left[X_o(0) - X_o(4)\right]$$

$$X_1(5) = W_N^1 \left[X_o(1) - X_o(5)\right]$$

$$X_1(6) = W_N^2 \left[X_o(2) - X_o(6)\right]$$

$$X_1(7) = W_N^3 \left[X_o(3) - X_o(7)\right]$$

The other column computations can be found in a similar manner using the transformation algorithm as shown in equation (6). The number of column computations will always be equal to M.

V. PARAMETERS

Selection of FFT Parameters:

A. The number of input samples (N) must be a power to two.

B. The sampling rate (f_s) must be greater than twice the highest possible frequency in the spectrum (f_h) to avoid aliasing.

C. The period of the time function (t_p) is chosen to be longer than the time length of the signal to prevent overlapping.

D. The desired frequency resolution (F) is

$$F = \frac{1}{t_p} \tag{7}$$

E. The only way to select both t_p and f_s independently is to use the relationship.

$$N \geq \frac{2 f_h}{F} \tag{8}$$

VI. RADIX-2 FFT COMPUTER PROGRAM

A computer program which uses the Radix-2 FFT algorithm developed in this report and containing a binary unscrambling algorithm is shown in the Appendix.

The computer program is written as a subroutine. There are three inputs necessary to run the FFT subroutine.

A. N

B. M

C. The input samples.

The input samples (N) are placed into two arrays as listed below.

P (N, 1)	Real part of sample
P (N, 2)	Imaginary part of sample

If the input samples are all real numbers, F (N,2) would be zero filled.

In order to insure reliable FFT results, care should be taken in selecting the input data sampling rate and sampling duration.

The output of the FFT subroutine is contained in the two arrays P (N, 1) and P (N, 2). The series of magnitudes found from these arrays is the discrete Fourier series equally spaced F units apart. Recalling that a Fourier series separates a periodic function of period T into sinuisoidal components of frequency ω_0, $2\omega_0, \ldots, \eta\omega_0$, where $\omega_0 = 2\pi/T$ is the fundamental frequency and the other frequencies, $2\omega_0, \ldots, \eta\omega_0$, are the harmonics of ω_0, the output of the FFT subroutine is in fact a harmonic analysis where the output magnitudes are amplitudes of signal components are discrete frequency intervals.

VII. CHECKOUT

The FFT program was checked out by inputting samples from the function

$$f(t) = 5 \sin \omega_1 t + 10 \sin \omega_2 t$$

with

$\omega_1 = 6.28$ RAD/SEC and $\omega_2 = 31.42$ RAD/SEC

$f_1 = 1.0\ H_z$ and $f_2 = 5\ H_z$

The following FFT parameters were used:

$N = 1024$

$t_p = 2.0$ sec

$F = 0.5\ H_z$

The output of the FFT program is shown in Table 2. The table shows the expected results of a 5 degree signal at 1 Hz, and a 10 degree signal at 5 Hz. - which are the same results that would be obtained from a DFT.

TABLE 2. FFT Output For The Checkout Example

FREQ (Hz)	SPECTRUM MAGNITUDE (DEGREE)
0.0000000E+00	5.6377379E-05
0.5000000	7.5358449E-04
1.000000	5.000081
1.500000	9.6538331E-04
2.000000	1.4357937E-03
2.500000	2.4941278E-04
3.000000	1.7234393E-03
3.500000	1.2[illegible]00881E-03
4.000000	1.9575893E-03
4.500000	1.1196[illegible]62E-03
5.000000	9.999151
5.500000	9.7630627E-04
6.000000	5.1663197E-04
6.500000	1.0909615E-03
7.000000	3.2764091E-04
7.500000	9.0288481E-04
8.000000	7.0783962E-04
8.500000	7.4152061E-04
9.000000	1.0592474E-03
9.500000	3.6268751E-04
10.00000	1.7118506E-03

To check the effect of sampling rate and the length of record (t_p), Table 3 was constructed. A spectral analysis (FFT) of $f(t) = 10 \cos 20 \pi t$ was done for the cases $t_p = .1$, 1, and 4 sec. and f_s was varied until the minimum sampling frequency was found.

Table 3 shows that a minimum sampling frequency for the 10.0 Hz signal is 256 Hz for t_p equal to 1 and 4. The sampling rate at t_p equal to .1 is larger, but this is due to the discrete values that N must have which then causes a discrete jump in the sampling frequency as seen between $t_p = .1$ and 1 sec. The minimum sampling frequency produces an impulse at ω_0 and zeroes at all other frequencies. It should be noted that at lower sampling frequencies the impulse at ω_0 can be seen along with other low impulses at other frequencies.

TABLES 3. Minimum Sampling Times and Frequency for $f(t) = 10 \cos 20 \pi t$.

t_p (sec)	.1	1	4
f_s (Hz)	640	256	256
N	64	256	1024
Cycles of Input	1	10	40

The main sources of error for the FFT are sampling rate, quantization and round-off. The sampling rate error as given by Hopper and Newberry [3.] is

$$E_s \leq \frac{8\,\pi^2}{12\,K^2} \quad \text{percent} \tag{9}$$

where K is the number of samples taken on each cycle of f(t), and the quantization and round-off error as given by Welch [4.] is

$$E_q \leq 0.1 \quad \text{percent} \tag{10}$$

Using equation (9) with the data in Table 3., E_s is found to be zero for t_p equal to .1 sec,and 0.01 percent for t_p equal to 1 and 4 sec.

VIII. CONCLUSION

The FFT subroutine presented in this report gives good results in the frequency domain. The results are discrete and are nonexistant between adjacent frequency intervals.

Nothing is known between the discrete frequencies. For more resolution of frequency, the record lenght, t_p, must be increased. Care must be taken to ensure that ω_0 is one of the discrete frequencies calculated, and the sampling frequency is high enough to yield low error results.

Overall, the FFT subroutine is a very useful, fast computational algorithm which can be used with any digital system simulation when frequency spectrum processing is needed in the calculation of the system's frequency response.

REFERENCES

1. Stanley, William D., "Digital Signal Processing", Reston Publishing Company, 1975.

2. Oppenheim, Alan V., "Digital Signal Processing", Prentice-Hall, 1975.

3. Hooper and Newberry, "Some Applications and Limitations of the Fast Fourier Transform ", NASA Technical Memorandum, NASA TM X-53997.

4. Welch, P.D., "A Fixed-Point Fast Fourier Transform Error Analysis". Vol. AU-17, IEEE Trans. Audio and Electracoustics, June 1969, pp. 151-157.

APPENDIX

FFT SUBROUTINE
(VAX-11 FORTRAN)

```
SUBROUTINE FFT
COMMON C(2000)
EQUIVALENCE   (C( 900),IP9  )
EQUIVALENCE   (C( 901),IP8  )
DIMENSION  P(8192,2)
```

PARAMETERS

```
INPUT**********************FOURIER TRANSFORM************************
P - AN ARRAY DIMENSIONED MAX--P (8192,2)
             DIMENSIONED--P(IP9,2) USED FOR THE DATA
IP9 = N = A POWER OF 2 = THE NUMBER OF DATA POINTS
IP9 MUST BE A POWER OF 2

IP9 = 2**IP8            IP8 = M             MAX----8192 = 2**13

OUTPUT*********************FOURIER TRANSFORM***********************
P - THE TRANSFORMED DATA.  THIS MEANS THE INITIAL DATA IS DESTROYED BY
    THE ROUTINE AND REPLACED WITH THE TRANSFORMED DATA.

*****FFT*****

IQ1 = 2*IP9
Q6=3.141592654/IP9
 DO 30 I=1,IP8
 IQ1=IQ1/2
 IQ2=IQ1/2
 Q6=Q6*2.0
 JQ=IP9+1-IQ1
 DO 20 J=1,JQ,IQ1
 Q5=-Q6
 IP4=J-1
 DO 10 K=1,IQ2
 Q5=Q5+Q6
 IQ3=IP4+K
 IQ4=IQ3+IQ2
 Q7=COS(Q5)
 Q8=SIN(Q5)
 Q9=Q7*(P(IQ3,1)-P(IQ4,1))+Q8*(P(IQ4,2)-P(IQ3,2))
 Q8=Q7*(P(IQ3,2)-P(IQ4,2))+Q8*(P(IQ3,1)-P(IQ4,1))
 P(IQ3,1)=P(IQ3,1)+P(IQ4,1)
 P(IQ3,2)=P(IQ3,2)+PIQ4,2)
 P(IQ4,1)=Q9
 P(IQ4,2)=Q8
10   CONTINUE
20   CONTINUE
30   CONTINUE

DO  60 I=1, IP9
IQ3=I-1
IQ4=0.0
```

```
 DO 50 J=1,IP8
 IQ4=2*IQ4
 P4=FLOAT(IQ3)/2.0
 IF(INT(P4).EQ.P4)GOTO 40
 IQ4=IQ4+1
40   IQ3=INT(P4)
50   CONTINUE


 IQ4=IQ4+1
 IF(IQ4.LE.I)GOTO 60
 Q9=P(I,1)
 Q8=P(I,2)
 P(I,1)=P(IQ4,1)
 P(I,2)=P(IQ4,2)
 P(IQ4,1)=Q9
 P(IQ4,2)=Q8
60   CONTINUE


 DO 70 I=1,IP9
 P(I,1)=P(I,1)/IP9
 P(I,2)=P(I,2)/IP9
70   CONTINUE


 J=IP9
 K=IP9/2

 DO 80 I=2,K
 Q1=P(I,1)
 Q2=P(I,2)
 P(I,1)=P(J,1)
 P(I,2)=P(J,2)
 P(J,1)=Q1
 P(J,2)=Q2
 J=J-1
80   CONTINUE


PRINT OUT FREQ (FQFFT) AND MAG (FFTM) OF FFT ***************

 DO 90 I=1,30
 FFTM=SQRT((P(I,1)*2.0)**2+(P(I,2)*2.0)**2)
 PRINT*,FQFFT,FFTM
 FQFFT=FQFFT+1.0/(IP9*DTFFT)
90   CONTINUE
         END IF
 RETURN
21   FORMAT(2(1x,G14.6))
 END
```

DISTRIBUTION

	No. of Copies
US Army Material System Analysis Activity ATTN: AMXSY-MP Aberdeen Proving Ground, MD 21005-5071	1
ITT Research Institute ATTN: GACIAC 10 W. 35th Street Chicago, IL 60616	1
AMSMI-R, Dr. McCorkle	1
Dr. Rhoades	1
-RG	1
-RGC	15
-RGT	2
-RPT (Record Copy)	1
-RPR	5
-LP, Mr. Voigt	1
AMCPM-ADCC-D, Linda Beach	2

END

FILMED

12-85

DTIC