SHARE PROGRAM LIBRARY AGENCY

PROGRAM NUMBER

163002

University of Miami

1365 MEMORIAL DRIVE - CORAL GABLES, FLORIDA

(305) - 284-6257

CONTRIBUTED PROGRAM LIBRARY SUBM
(for IBM S/360, 1130 and 1800)

SHARE Program Library Agency
Triangle Universities Computation Center
P. O. Box 12076
Research Triangle Park, N. C. 27709

This form should be completed and submitted with the program package to PID at the address shown above. Standards and instructions for submitting programs are in your *User Group Reference Manual* or the *Contributed Program Submittal Standards Manual* available from PID.

1. Program Order Number *(to be filled in by PID)* . . . 360D-16.3.002
2. System Type *(machine)* . . . S/360
3. Search Key . . . /CONVERTS/ PULSE TEST /DATA/ /TO/ FREQUENCY RESPONSE DATA /VIA/ REAL-VALUED FAST FOURIER TRANSFORM
4. Name of Author *(if different than submitter's)* . . . Carlos Ray Dollar; Cecil L. Smith; Paul W. Murrill
5. Submitter's Name *(direct technical inquiries to)* . . . C. Ray Dollar
6. Submitter's Address . . . Process Computer Engineering, Dow Chemical Company, Freeport, TX 77541
7. Title of Program . . . Pulse Testing Via The Fast Fourier Transform
8. Submitter's User Group Affiliation Code and Installation Code . . . S VPI
9. Submitter's Own Program Identification and Suffix *(optional)* . . .
10. Primary Subject Code . . . 16.3
11. Secondary Subject Codes . . . 06.5 45.1 10.0
12. Operating or Monitor System Required . . . SEE ABSTRACT
13. New or Revision Code *(if revision, show prior Program Order Number in item 1)* . . . N
14. Year Completed . . . 69
15. Date of Submittal . . . 101769
16. Documentation *(number of original pages submitted)* . . . 12
17. Abstract *(should contain sufficient information for a reader to determine the value of the program). Listed on the reverse side of this form are subjects which may serve as a guide for a descriptive abstract.*

CONTRIBUTED PROGRAM LIBRARY SUBMITTAL FORM

Subject Guide

a. Purpose
b. Programming Language used
c. Version and modification level or release number of IBM Programming System used, or program order number for non-IBM authored program used
d. Field of application
e. Type of routine (main program, subroutine, etc.)
f. Specific description of machine requirements
g. Engineering Changes (EC) level of equipment (if pertinent)

ABSTRACT

SUBROUTINE REFFT ACCOMPLISHES A FAST, ACCURATE TRANSFORMATION OF TIME DOMAIN PULSE RESPONSE DATA TO FREQUENCY RESPONSE DATA USING THE REAL-VALUED FAST FOURIER TRANSFORM. THE SUBROUTINE IS WRITTEN IN FORTRAN IV AND HAS BEEN COMPILED AND TESTED ON A 7040, A S/360 MODEL 50, AND A S/360 MODEL 65. GIVEN TIME DOMAIN INPUT AND OUTPUT PULSE DATA THE SUBROUTINE WILL CALCULATE AND PRINT MAGNITUDES, PHASE ANGLES AND ASSOCIATED FREQUENCIES

PROGRAMMING SYSTEMS- Written in FORTRAN IV

MINIMUM SYSTEM REQUIREMENTS- A minimum of 16K of core storage is required and will run on a S/360 Model 30.

(Please attach additional pages if necessary) . . . . **Total pages attached** _____

Permission to Publish

"I hereby give anyone permission to reprint, reproduce, and distribute this program to anyone else."

(18) **Signature of Submitter and Date** Carlos Ray Dollar 10/17/69

(19) **Signature of Installation Addressee** [signature]

T4SF

TABLE OF CONTENTS

Card Deck Key

Deck #1	Fortran Source Deck, sequence 0001 through 0177 in cc 77-80; RFFT in cc 73-76, 177 cards
Deck #2	Example Problem (calling program and data subroutine), sequence 0001 through 0061 in cc 77-80; EP in cc 75-76, 61 cards

The following is a description of the implementation and use of subroutine REFFT.

PURPOSE

REFFT is a subroutine which will quickly convert pulse response time data to frequency response data.

DESCRIPTION

This subroutine uses the real-valued Fast Fourier Transform for converting pulse response data from the time domain to the frequency domain. The ratio of the time required for calculating Fourier transforms via the Fast Fourier Transform to the time required by conventional techniques is approximately $\frac{\log_2 N}{N}$, where N is the number of data points. As N increases, the reduction in time required quickly becomes very large. The real-valued Fast Fourier Transform is a Fast Fourier Transform technique written especially for the case where the input is restricted to real numbers and it takes advantage of this to eliminate unnecessary calculations. The technique is described in detail in "A Fast Fourier Transform Algorithm for Real-Valued Series," by G. D. Bergland in Communications of the ACM, Vol. 11, No. 10, pp. 703-710, October, 1968.

Subroutine REFFT has been written specifically for converting pulse response data and computes the input and output pulse transforms simultaneously, thereby effecting additional economies in calculation. The accuracy of REFFT is at least as good as that of conventional techniques and in some cases it is better than conventional techniques. The printed output provided by REFFT is phase angles, amplitude ratios and associated frequencies.

IMPLEMENTATION

REFFT assumes that the same constant time increment, ST, between data points is used for both the input and output pulses. REFFT must be provided with NX, NY, ST, X, Y from the main program where

NX = number of data points in X array
NY = number of data points in Y array
ST = time increment between data points
X = input pulse data array
Y = output pulse data array

These values enter REFFT in a COMMON statement and are used to calculate and print out phase angles, amplitude ratios and frequencies where

PHA = phase angle, degrees
G = amplitude ratio
W = frequency corresponding to G and PHA, radians/time unit

The frequencies will range between 0.0 and 1/2 the sampling frequency (sampling frequency = $2\pi/ST$). The number of frequencies will be equal to one plus one-half the smallest power of 2 which is greater than or equal to NY. REFFT is designed to accommodate an NY of up to 1024.

MODIFICATIONS

Changing the maximum NY which the program will handle is relatively simple. Let MN be the new maximum NY. MN must be a power of 2($1024 = 2^{10}$) which is greater than or equal to any NY desired. The memory allocations in DIMENSION and COMMON must be as follows:

DIMENSION KE($\frac{MN}{2}$), WC($\frac{MN}{4}$), WS($\frac{MN}{4}$), PHA($\frac{MN}{2}$ + 1), G($\frac{MN}{2}$ + 1)

Dimension AR($\frac{MN}{2}$ + 1), AI($\frac{MN}{2}$ + 1), W($\frac{MN}{2}$ + 1)

COMMON NX, NY, ST, X(MN), Y(MN)

In addition in subroutine REFFT the statement

```
DO 7 M = 1,10
```

must be changed to

```
DO 7 M = 1,P
```

where $2^P = MN$

Increasing the frequency resolution involves only two changes. The program searches to find NN, the smallest power of 2 greater than or equal to NY, and outputs a number of frequencies equal to $\frac{NN}{2} + 1$. To increase the resolution, then, in REFFT, statement 1 can be changed to read, for example,

```
1 NN = 2 * I
```

(NN must be $\leq$ MN)

If still greater resolution is desired then multiply I by a larger power of 2. When this modification is made the following statement should be added immediately following statement 1:

```
M = M + K
```

where 2^K = the power of 2 multiplied times I in statement 1

Should the real and imaginary values of the complex point represented by the amplitude ratio G and the phase angle PHA be desired, they are readily available. The complex point $G(j\omega)$ is calculated as

$$G(j\omega) = AR + j\,AI$$

where AR = real value of $G(j\omega)$

AI = imaginary value of $G(j\omega)$

AR and AI are available in the program and can be outputted as desired.

EXAMPLE

Given data points for the input pulse, X(t), and the output pulse, Y(t), from a pulse response test on a system it is desired to convert this data to frequency response data, i.e., phase angles, amplitude ratios, and frequencies. The data, X(t) and Y(t), could be read into the program from data cards, but in this example the data will be generated in Subroutine DATA. For this case NX = NY but this is not necessary and involved no changes in REFFT. The pulse input used has the equations

$$X(t) = \frac{8t^3}{T_p} \qquad \text{for} \qquad 0 \leq t \leq \frac{T_p}{2}$$

$$X(t) = 8\left(1 - \frac{t}{T_p^3}\right) \qquad \text{for} \qquad \frac{T_p}{2} < t \leq T_p$$

where T_p = pulse duration

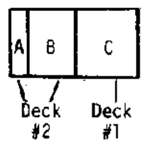
The system is a first order lag with a time constant of 0.1.

The deck arrangement for running this example problem is shown on page 9. Subroutine DATA generates all of the input data.

Pages 10 and 11 give the output for an ST (time increment between data points) of 0.03 and 0.06 respectively. The subroutine DATA which is provided specifies ST = 0.03 (first statement following the COMMON statement in subroutine DATA). This may be changed to ST = 0.06 to compare results with the provided ST = 0.06 results.

Flow Diagram for Example Problem

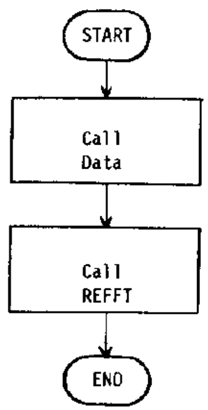
Deck Arrangement: Front



A. Main Program (Calling Program)

B. Subroutine Data

C. Subroutine REFFT



EXAMPLE PROBLEM OUTPUT

ST = 0.03

W	G	Angle
0.0	1.0012894	0.0
3.2725	0.9518750	-18.0712
6.5450	0.8386129	-33.1127
9.8175	0.7160900	-44.3395
13.0900	0.6105056	-52.4363
16.3625	0.5261577	-58.2988
19.6349	0.4596455	-62.6254
22.9074	0.4068800	-65.8939
26.1799	0.3648164	-68.3844
29.4524	0.3308951	-70.2747
32.7249	0.3030094	-71.7230
35.9974	0.2798783	-72.8183
39.2699	0.2606833	-73.5720
42.5424	0.2446392	-74.0257
45.8149	0.2311110	-74.2342
49.0874	0.2197269	-74.1845
52.3599	0.2102307	-73.8450
55.6324	0.2024853	-73.2313
58.9048	0.1962180	-72.3312
62.1773	0.1911980	-71.1032
65.4498	0.1875842	-69.5077
68.7223	0.1854712	-67.4825
71.9948	0.1844578	-64.9807
75.2673	0.1845198	-61.9969
78.5398	0.1862038	-58.3912
81.8123	0.1893615	-53.9898
85.0848	0.1933089	-48.8766
88.3573	0.1983089	-43.0023
91.6298	0.2046027	-36.0491
94.9023	0.2108877	-28.0272
98.1747	0.2160779	-19.3452
101.4472	0.2200236	-10.0206
104.7197	0.2216268	0.0

EXAMPLE PROBLEM OUTPUT

ST = 0.06

W	G	PHA
0.0	1.0039845	0.0
3.2725	0.9546932	-18.1188
6.5450	0.8419061	-33.2373
9.8175	0.7204924	-44.6269
13.0900	0.6170442	-53.0923
16.3624	0.5359110	-59.6859
19.6349	0.4728857	-65.1309
22.9074	0.4236732	-69.9493
26.1799	0.3863571	-74.8060
29.4524	0.3589668	-80.2706
32.7249	0.3388029	-86.5703
35.9974	0.3248897	-94.1968
39.2699	0.3182657	-104.3540
42.5424	0.3185146	-117.9323
45.8149	0.3229568	-134.8878
49.0874	0.3291715	-155.6068
52.3599	-0.3326269	180.0000

Acknowledgement of Support

This work was supported in part by Project Themis Contract No. F44620-68-C-0021 administered for the U. S. Department of Defense by the U. S. Air Force Office of Scientific Research. Additional support was provided by the National Science Foundation.