AN ABSTRACT OF THE THESIS OF


Title: Two Dimensional Recursive Fast Fourier Transform

Abstract approved: Redacted for Privacy

Wojciech J. Kolodziej

The purpose of this thesis is to extend one dimensional recursive fast Fourier transform (1-D RFFT) into two dimensions, by which the two dimensional signal can be processed recursively and the desired spectrum can be obtained.

Compared with FFT, RFFT has the following advantages: (1) No requirement that the number of input data should be equal to the number of frequencies. (2) Fourier transform can start before all the data are obtained. The latter makes this algorithm suitable for on-line spectral analysis.

In this thesis, a two dimensional RFFT algorithm is proved mathematically, and an application to signal detection is demonstrated. Also, this algorithm can be applied to image processing and other two dimensional signal processing problems.
TWO DIMENSIONAL RECURSIVE FAST FOURIER TRANSFORM

by

ZHU, FEIHONG

A THESIS

submitted to

Oregon State University

in partial fulfilment of
the requirements for the

degree of

Master of Science

Completed March 9, 1987

Commencement June, 1987
PROPOSED:

Redacted for Privacy

Dr. W. J. Kolodziej, Department of Electrical & Computer Engineering, in charge of major

Redacted for Privacy

Dr. S. J. J. Owen, Chairman of Department of Electrical & Computer Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented March 9, 1987
ACKNOWLEDGEMENT

Special gratitude is due to my major professor Dr. Wojciech J. Kolodziej for his great help and support during the period of doing my thesis.

I would also like to thank the other members of my committee, Dr. Herzog J. H., Dr. Mohler R.R. and Dr. Fredericks William J., for their interest and helpful suggestions.

I especially thank my loving wife, Xu Yan, your constant support and sacrifice made this endeavor a reality. I am very proud and fortunate to have a wife like you.

A big thanks goes to Mr. Fred M. Quick and Mrs. Evelyn A. Quick whose support and never-ending encouragement made this thesis possible.

My mother, Mi Xurong, my father Zhu Peiji and my aunt, Zhu Wanying, deserve special thanks for your support, encouragement and love you have given to me throughout my life.

Also I wish to express deep gratitude to my brother, Zhu Changhong, and other members of my family for their continuous understanding and support.
# Table of Contents

## I Introduction

## II Literature Review

1. Development of FFT

2. Recursive Algorithm of DFT

## III Two Dimensional RFFT

1. Recursive Algorithm of RFFT

2. Fast Algorithm of RFFT

## IV Experimental Results

## V Conclusion and Possible Applications

## VI Bibliography
<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. 8-point Butterfly Diagram of General FFT</td>
<td>8</td>
</tr>
<tr>
<td>2. Two Dimensional Data Array</td>
<td>18</td>
</tr>
<tr>
<td>3. 8-point Butterfly Diagram of Revised FFT</td>
<td>32</td>
</tr>
<tr>
<td>4. Diagram of RFFT Calculation</td>
<td>35</td>
</tr>
</tbody>
</table>
| 5. Experiment1: $X(u,v) = \exp(-u-v)$ \( u \geq 0, v \geq 0 \)  
N1=128 (H=4, N=32)  
N2=128  
\( \Delta u=0.01 \)  
\( \Delta v=0.01 \)  
A1=128  
A2=128 | 47 |
| 6. Experiment1: $X(u,v) = \exp(-u-v)$ \( u \geq 0, v \geq 0 \)  
N1=256 (H=8, N=32)  
N2=250  
\( \Delta u=0.01 \)  
\( \Delta v=0.01 \)  
A1=128  
A2=128 | 48 |
| 7. Experiment2: $X(u,v) =  
\begin{cases} 
\exp(-0.1v+u^2/4v)/v^{1/2} & u \geq 1, v \geq 0 \\
0 & u < 1, v < 0 
\end{cases}$  
N1=64 (H=1, N=64)  
N2=50  
\( \Delta u=0.1 \)  
\( \Delta v=0.1 \)  
A1=128  
A2=128 | 49 |
| 8. Experiment2: $X(u,v) =  
\begin{cases} 
\exp(-0.1v+u^2/4v)/v^{1/2} & u \geq 1, v \geq 0 \\
0 & u < 1, v < 0 
\end{cases}$  
N1=64 (H=1, N=64)  
N2=100  
\( \Delta u=0.1 \)  
\( \Delta v=0.1 \)  
A1=128  
A2=128 | 49 |
| 9. Experiment2: $X(u,v) =  
\begin{cases} 
\exp(-0.1v+u^2/4v)/v^{1/2} & u \geq 1, v \geq 0 \\
0 & u < 1, v < 0 
\end{cases}$  
N1=128 (H=1, N=128)  
N2=128  
\( \Delta u=0.1 \)  
\( \Delta v=0.1 \)  
A1=128  
A2=128 | 50 |
| 10. Experiment2: $X(u,v) =  
\begin{cases} 
\exp(-0.1v+u^2/4v)/v^{1/2} & u \geq 1, v \geq 0 \\
0 & u < 1, v < 0 
\end{cases}$  
N1=128 (H=2, N=64)  
N2=128  
\( \Delta u=0.1 \)  
\( \Delta v=0.1 \)  
A1=128  
A2=128 | 50 |
Figure

11. Experiment2: \( X(u,v) = \begin{cases} \exp(-0.1v+u^2/4v)/v^{1/2} & u \geq 1, v \geq 0 \\ 0 & u < 1, v < 0 \end{cases} \)
\( N_1 = 192 \) (\( H = 3, N = 64 \)) \( N_2 = 200 \)
\( \Delta u = 0.1 \) \( \Delta v = 0.1 \)
\( A_1 = 128 \) \( A_2 = 128 \)

12. Experiment2: \( X(u,v) = \begin{cases} \exp(-0.1v+u^2/4v)/v^{1/2} & u \geq 1, v \geq 0 \\ 0 & u < 1, v < 0 \end{cases} \)
\( N_1 = 256 \) (\( H = 2, N = 128 \)) \( N_2 = 200 \)
\( \Delta u = 0.1 \) \( \Delta v = 0.1 \)
\( A_1 = 128 \) \( A_2 = 128 \)

13. Experiment2: \( X(u,v) = \begin{cases} \exp(-0.1v+u^2/4v)/v^{1/2} & u \geq 1, v \geq 0 \\ 0 & u < 1, v < 0 \end{cases} \)
\( N_1 = 384 \) (\( H = 3, N = 128 \)) \( N_2 = 350 \)
\( \Delta u = 0.1 \) \( \Delta v = 0.1 \)
\( A_1 = 128 \) \( A_2 = 128 \)

14. Experiment3: Spectrum of Two Dimensional White Gaussian Noise \( n(u,v) \)

15. Experiment3: \( X(u,v) = \sin(\pi(u-4)-\pi(v-4))U(u-4,v-4) + n(u,v)U(u,v) \)
\( N_1 = 112 \) (\( H = 7, N = 16 \)) \( N_2 = 112 \)
\( \Delta u = 0.125 \) \( \Delta v = 0.125 \)
\( A_1 = 64 \) \( A_2 = 64 \)

16. Experiment3: \( X(u,v) = \sin(\pi(u-4)-\pi(v-4))U(u-4,v-4) + n(u,v)U(u,v) \)
\( N_1 = 128 \) (\( H = 4, N = 32 \)) \( N_2 = 128 \)
\( \Delta u = 0.125 \) \( \Delta v = 0.125 \)
\( A_1 = 64 \) \( A_2 = 64 \)

17. Experiment3: \( X(u,v) = \sin(\pi(u-4)-\pi(v-4))U(u-4,v-4) + n(u,v)U(u,v) \)
\( N_1 = 160 \) (\( H = 5, N = 32 \)) \( N_2 = 150 \)
\( \Delta u = 0.125 \) \( \Delta v = 0.125 \)
\( A_1 = 64 \) \( A_2 = 64 \)

18. Experiment3: \( X(u,v) = 0.1\sin(\pi(u-4)-\pi(v-4))U(u-4,v-4) + n(u,v)U(u,v) \)
\( N_1 = 640 \) (\( H = 10, N = 64 \)) \( N_2 = 640 \)
\( \Delta u = 0.125 \) \( \Delta v = 0.125 \)
\( A_1 = 64 \) \( A_2 = 64 \)
LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Spectrum Analysis with Transfer Function&lt;br&gt;$F(w_1,w_2)=1/((jw_1+1)(jw_2+1))$</td>
<td>56</td>
</tr>
<tr>
<td>2. Spectrum Analysis with DFT&lt;br&gt;(N1=28, N2=30)</td>
<td>57</td>
</tr>
<tr>
<td>3. Spectrum Analysis with RFFT&lt;br&gt;(N1=28, N2=30)</td>
<td>58</td>
</tr>
<tr>
<td>4. Spectrum Analysis with FFT&lt;br&gt;(N1=16, N2=16)</td>
<td>59</td>
</tr>
<tr>
<td>5. Spectrum Analysis with RFFT&lt;br&gt;(N1=8, N2=10)</td>
<td>60</td>
</tr>
<tr>
<td>6. Spectrum Analysis with RFFT&lt;br&gt;(N1=16, N2=16)</td>
<td>61</td>
</tr>
<tr>
<td>7. Spectrum Analysis with RFFT&lt;br&gt;(N1=20, N2=25)</td>
<td>62</td>
</tr>
<tr>
<td>8. Spectrum Analysis with RFFT&lt;br&gt;(N1=56, N2=60)</td>
<td>63</td>
</tr>
</tbody>
</table>
CHAPTER I

INTRODUCTION

Since the appearance of fast Fourier transform (FFT) (Cooley/Tukey, 1965) (Cooley/Cochran, 1967), many researchers have worked in this field and invented a variety of FFT algorithm implementations. FFT is an efficient method of computing discrete Fourier transform (DFT), but it requires that all the data sequence be obtained before the start of transformation. This limitation makes FFT difficult to be used for on-line applications because sometimes the duration of signal is unknown. Therefore the search of recursive algorithms of Fourier transform is attractive.

For one dimensional DFT, Ahmed (Ahmed/Natarajan/Rao, 1973) presented a recursive algorithm which was derived from the "mirror image" of the signal. By processing the input data one by one, this method can be used for on-line spectral identification. Hostetter (Hostetter, 1980) constructed another recursive algorithm based on the observer concept of control theory.
His first paper prompts more efforts in this area (Hostetter, 1983) (Bitmead, 1982). These approaches have recursive property, but the transform speed is low because the data are processed one by one. Therefore, a question is raised: how to find a method which has both fast and recursive properties. For this problem, a moderate method, recursive fast Fourier transform (RFFT), was presented (Zhu, 1985) (Zhu/Wang, 1986). This method is developed from Ahmed's method, but the input data are processed segment by segment recursively, as opposed to one by one, and a revised fast algorithm is used in processing every segment.

Compared with FFT, this method has the following advantages:

(1) Flexibility: The number of input data in every segment may not equal the number of frequencies.

(2) Recursive property: Due to no limitation on recursive steps, the desired spectrum can be obtained as the input data increase.

Compared with other recursive methods mentioned above, this RFFT approach has higher transform speed because a revised fast algorithm is used in every recursive computation. The main idea of the algorithm is to transform the new data segment to frequencies by means
of revised FFT algorithm and then to "repair" the old spectrum which is obtained by previous data. Although the total computational operations of RFFT are more than those of FFT, its transform speed may not be low because the part of computation can be finished between the sampling intervals.

For two dimensional DFT, the common methods used so far are still FFT type. The purpose of this thesis is to extend one dimensional RFFT to two dimensional case because recursive algorithm is also attractive for two dimensional signal processing.

The organization of this thesis is as follows. In chapter II, a literature review is given, which introduces the development of fast Fourier transform, recursive Fourier transform and other related algorithms for the computation of DFT. Chapter III presents the algorithm of two dimensional recursive fast Fourier transform. This algorithm consists of two parts: the recursive algorithm and the revised fast algorithm. Chapter IV describes numerical experiment results, including a simulation of signal detection by RFFT. In chapter V, the conclusion and discussion of possible applications of RFFT algorithm are given.
CHAPTER II

LITERATURE REVIEW

1. DEVELOPMENT OF FFT

Fourier transform is an essential analysis method in many scientific and engineering fields. One of the reasons that Fourier transform have such wide-ranging applications is because of the existence of powerful digital computers and efficient algorithms for computing discrete Fourier transform.

The definition of DFT is given by

$$F(k) = \sum_{n=0}^{N-1} x(n)W^{kn}, \quad k=0,1,...,N-1,$$

(2-1)

where \( W = \exp(-j2\pi/N) \) and \( x(n), \ n=0,1,2,...,N-1, \) is the sample sequence of a time signal.
Inverse discrete Fourier transform (IDFT) is given by

\[ x(n) = (1/N) \sum_{n=0}^{N-1} F(k)W^{kn}, \quad n=0,1,...,N-1. \quad (2-2) \]

To indicate the importance of efficient computational procedures, it is necessary to consider the direct calculation of DFT equations. Since \( x(n) \) may be complex, it can be written as follows:

\[ F(k) = \sum_{n=0}^{N-1} \{ \text{Re}[x(n)]\text{Re}[W^{kn}] - \text{Im}[x(n)]\text{Im}[W^{kn}] \} \]

\[ + j(\text{Re}[x(n)]\text{Im}[W^{kn}]+\text{Im}[x(n)]\text{Re}[W^{kn}]), \]

\[ k=0,1,...,N-1. \quad (2-3) \]

From Eq.(2-3), it can be noticed that for each value of \( k \), the direct computation of \( F(k) \) requires \( 4N \) real multiplications and \( 4N-2 \) real additions. Since \( F(k) \) must be computed for \( N \) different values of \( k \), the direct computation of DFT for a sequence \( x(n) \) requires \( 4N^2 \) real multiplications and \( (4N-2)N \) real additions. It is evident that the number of arithmetic operations required to compute DFT by the direct method becomes very large for the large values of \( N \) because the amount of the computation is approximately proportional to \( N^2 \). For this reason, computational procedures that reduce the number of the arithmetic operations are of considerable interest.

Most algorithms which improve the efficiency of DFT
computation depend on one or both of the following special properties of the quantity $W$:

1. $W^{k(n-n/^2)} (W^{kn})^*$,
2. $W^{kn} = W^{k(n+n)} = W^1$,

where $(W)^*$ is the complex conjugate of $W$.

Computational algorithms that use both (1) the symmetry and (2) the periodicity of the sequence $W$ were known long before the appearance of high-speed digital computation for Fourier transform (Colley/Cochran, 1967). However this is not of great importance for the small values of $N$ that are feasible for hand computations. The possibility of greatly reduced computation was generally overlooked until about 1965, when Cooley and Tukey (Cooley/Tukey, 1965) published the famous algorithm of fast Fourier transform. This algorithm requires much less computational effort and can be applied when $N$ is a composite number (i.e., $N$ is a product of two or more integers). The publication of this paper makes it possible to apply discrete Fourier transform to signal processing and results in the discovery of a variety of FFT algorithms.

The fundamental principle that FFT algorithms are based on is that of decomposing the computations of DFT sequence of length $N$ into successively smaller DFT's. The
manner in which the principle is implemented leads to many different algorithms, all with some improvements in computational speed. For the sake of comparison with RFFT, two basic classes of FFT are introduced: decimation-in-time algorithm and decimation-in-frequency algorithm.

To achieve the dramatic increase in efficiency which has been mentioned, it is necessary to decompose DFT computation into successively smaller DFT computations. In this process both the properties of the symmetry and the periodicity of complex exponential $W^k_n = \exp(-j(2\pi/N)kn)$ are used. Algorithms where the decomposition is based on decomposing the sequence $x(n)$, into successively smaller subsequences, are called decimation-in-time algorithms. They are originally discovered by Cooley and Tukey (Cooley/Tukey, 1965).

Alternatively, the computational methods can be considered dividing the output sequence $F(k)$ into smaller and smaller subsequences in the same manner. The class of FFT algorithms based on this procedure is commonly referred to as decimation-in-frequency (Oppenheim/Schafer, 1975).
The following is a flow chart of the decimation-in-frequency algorithm (N=8).

8-point Butterfly Diagram of General FFT

In Fig. 1, DFT is computed by first forming the sequence \( X_1(n) \) with \( X_1(n) = X(n) + X(n+N/2) \) and \( X_1(n+N/2) = X(n) - X(n+N/2) \), \( n=0,1,2,3 \), which results in two \( N/2 \)-point DFTs, then computing \( X_2(n) \) in the same manner, resulting in four \( N/4 \)-point DFTs which reduce the computation to the two-point DFTs and can be implemented by addition and subtraction. The computation of FFT
requires \( \frac{N}{2} \log_2 N \) complex multiplications and \( N \log_2 N \) complex additions. Thus the total computation is much less than that of direct computation of DFT for a large \( N \).

Based on these principles, several algorithms have been developed to speed up the transform even further. The algorithm proposed in 1982 (Preuss, 1982) reduces the number of multiplications and shows that the DFT of a complex sequence can be computed from the DFT of four real sequences which satisfies the appropriate symmetry conditions. Another algorithm is based on the fact that a radix-2 (i.e., \( N \) is a power of 2) algorithm diagram can be transformed into a radix-4 algorithm diagram. This process requires fewer multiplications and additions (Duhamel/Hollmann, 1984). Also, an approach is proposed for the implementation of radix-2 FFT algorithm in multiprocessors to increase the efficiency of the transform (Bhuyan/Agrawal, 1983). The methods discussed above are radix-2 algorithms, the decomposition of which leads to a highly efficient computation of DFT. However, in some cases it may not be possible to choose \( N \) to be a power of 2. Thus it is necessary to consider the application in the case where \( N \) is a product of factors that are not all necessarily equal to 2 (Gentleman, 1966) (Singleton, 1969).

For the Fourier transform at arbitrary frequencies,
an algorithm proposed is based on the fact that the Fourier transform at an arbitrary frequency can be expressed as a weighted sum of its DFT coefficients. The method retains the computational order of FFT but allows the flexibility of choosing arbitrary frequencies for a uniformly sampled signal (Sudhakar, 1981).

Besides FFT, two additional methods have been developed recently for the computation of DFT: the polynomial transforms (Pei/Wu, 1981) (Wu/Pei, 1984) (Nussbaumer, 1979) and estimator methods (Charles, 1984) (Charles, 1982). The underlying idea of the estimator methods is to approximate the designed estimator by iteratively solving the corresponding matrix equation, which makes it computationally feasible for the large system equations.

For two dimensional DFT, general method is by using one dimensional FFT repeatedly. The computational procedures consist of three steps: First, each row in the array is transformed with one dimensional FFT. Second, the array is transposed. Finally, the row transform process is repeated. This three-step procedure yields the transposition of the two dimensional transform array (Mitra, 1978). In addition, the contribution of Harris (Harris et al, 1977) described a calculation approach avoiding matrix transposition for the two dimensional FFT,
by means of decomposing the two dimensional transform into successively smaller two dimensional transforms. Mersereau proposed yet another transformation procedure (Mersereau, 1974) to change two dimensional sequences into one dimensional ones. Using the mapping method, several two dimensional problems can be solved by one dimensional techniques.

In contrast to one dimensional DFT, the computations required by two dimensional DFT may place severe demands on even the largest computers. This is because of the large size data arrays usually encountered in practical applications. For this problem, Eklundh presented a fast computer method for matrix transposition. This method enables applications of FFT to the matrices, the size of which exceeds available main storage (Eklundh, 1972). A similar approach was proposed later for the transposition of non-square matrices (Schumann, 1973). Another algorithm presented by Twogood and Ekstrom is a more efficient extension of Eklundh's basic method (only two rows of the matrix are resident in the primary storage at any one time) (Twogood/Ekstrom, 1976). Also, the algorithms developed by Onoe (Onoe, 1975), Dellotto and Dotti (Dellotto/Dotti, 1975), Hinton and Saleh (Hinton/Saleh, 1984) are worth being mentioned here.
2. RECURSIVE ALGORITHM OF DFT

Although FFT is an efficient method of Fourier transformation, it requires that the number of time sequence elements be equal to the number of frequencies and the computation begins only after all the time sequence is obtained. In order to process signals on-line, the recursive computation of DFT has been also developed in recent years. Ahmed, Natarajan and Rao presented (Ahmed/Natarjan/Rao, 1973) an algorithm derived from the discrete Fourier transform formula by using the "mirror image" of time-sample series. The main recursive formula of the algorithm is

$$Z(w,s) = L(w)Z(w,s-1) + BX(s),\quad s=0,1,2,\ldots,N-1,$$

where $X(s)$, $s=0,1,\ldots,N-1$, is the sample sequence of a
time signal, \( w \) is the frequency, and

\[
Z(w,-1) = \begin{pmatrix} 0 \\ B \\ 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad L(w) = \begin{pmatrix} \cos \omega t & -\sin \omega t \\ \sin \omega t & \cos \omega t \end{pmatrix},
\]

\( \Delta t \) is the sample interval.

This method provides a simple means of generating frequency amplitude plots of Fourier power and phase spectra recursively, which can be used in the spectral analysis of the signals whose duration is unknown. It was emphasized that FFT would be the most efficient way to compute spectra if the following constraints were satisfied:

i) The number of data \( N \) equals the number of frequencies \( A \), i.e., \( N = A \), and \( N = 2^n \), where \( n \) is a positive integer.

ii) \( w = 2\pi k / \Delta t A \), \( k = 0, 1, 2, \ldots, (A-1) \).

However, using Eq. (2-4), these constraints could be removed. Another recursive DFT algorithm proposed by Hostetter (Hostetter, 1980) is based on the concept of the observer of control theory. The paper is developed from the fact that a linear, time invariant homogeneous differential equation with the characteristic polynomial

\[
s(s^2 + (2\pi/T)^2)(s^2 + (4\pi/T)^2) \ldots (s^2 + (2N\pi/T)^2),
\]

has a general solution of the form
\[ y(t) = d_0 + \sum_{n=1}^{N} (a_n \cos(2\pi nt/T) + b_n \sin(2\pi nt/T)), \quad (2-5) \]

where \( d_0, a_n \) and \( b_n \) are the Fourier coefficients.

It is shown that the state variable representation for this homogeneous system is of the following form:

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\vdots \\
\dot{x}_{2N-1} \\
\dot{x}_{2N}
\end{bmatrix} =
\begin{bmatrix}
0 & 1 & 0 & 0 & \ldots & 0 & 0 \\
-(2\pi/T)^2 & 0 & 0 & 0 & \ldots & 0 & 0 \\
0 & 0 & 0 & 1 & \ldots & 0 & 0 \\
0 & 0 & -(4\pi/T)^2 & 0 & \ldots & 0 & 0 \\
& \ddots & & \ddots & \ddots & \ddots & \ddots \\
& & \ddots & & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\
0 & 0 & 0 & 0 & -(2N\pi/T)^2 & 0 & \ldots & 0
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
\vdots \\
x_{2N-1} \\
x_{2N}
\end{bmatrix}
\]

\[ = AX, \quad (2-6) \]

\[ Y = [1 \ 0 \ 1 \ 0 \ \ldots \ 1 \ 0 \ 1]X = C^T X. \quad (2-7) \]

From the viewpoint of this representation, deciding the coefficients of DFT is equivalent to the determination of the system initial conditions (i.e., to the classical deterministic observer problem).

Also, Hostetter proposed a recursive technique for Fourier spectral analysis, when the samples were unevenly spaced in time (Hostetter, 1983), and an approach to the inverse DFT (Hostetter, 1984), both of which are based on the observer of state variables. The latter gives a recursive solution for general discrete linear transform
and shows that DFT is a special case of this algorithm.

In 1982, Bitmead (Bitmead, 1982) published a paper, the aim of which is to demonstrate that the approach by Hostetter yields the final results essentially equivalent to standard signal processing methods. He makes explicitly the connections between them and points out the possible extensions.

In addition, Stuller (Stuller, 1982) presented an approach for recursive discrete transform with respect to arbitrary transform bases. He relates the generalized transform to recursive discrete Fourier transform and points out that it is a special case of this generalized transformation.

As discussed above, every method has its own advantages, but slow computation speed is their common disadvantage although the recursive method is attractive to on-line applications. And for the Hostetter's method, practical realizations limit its applications due to a high dimension of observer needed for the large number of the coefficients of the Fourier transform. To overcome these disadvantages and keep the recursive property, an algorithm "Recursive Fast Fourier Transformation" (RFFT) of one dimension has been proposed (Zhu, 1985) (Zhu/Wang, 1986). This algorithm combines the advantages of FFT and
the recursive computation of DFT. By processing all data segment by segment recursively and using the revised fast algorithm for every segment, this algorithm has high transform speed and can be used for on-line applications.

This thesis is focused on two dimensional recursive fast Fourier transform (2-D RFFT), which is developed from one dimensional RFFT. In addition, an application to signal detection is given.
CHAPTER III

TWO DIMENSIONAL RFFT

1. RECURSIVE ALGORITHM OF RFFT

The main idea of the two dimensional recursive fast Fourier transform is to transfer the k-th row data into the one dimensional spectrum by means of one dimensional revised FFT or RFFT and then use the recursive algorithm to "repair" the old spectrum which is obtained from previous data (see Fig.2). This process is repeated until all the data are transformed or the desired spectrum is obtained.

Next, the construction of the recursive algorithm will be discussed according to this idea.
DEFINITION 1:
A two dimensional discrete signal sequence is defined as follows:

\[ x^*(m_1,m_2) = \sum_{m_2} \sum_{m_1} X(u,v) \delta(u-m_1\Delta u) \delta(v-m_2\Delta v), \quad (3-1) \]

\[ m_1=0,1,2,...,N1-1, \quad m_2=0,1,2,...,N2-1, \]

where, \( X(u,v) \) is a two dimensional signal function, \( \Delta u \) and \( \Delta v \) are the sample intervals along horizontal and vertical directions respectively and \( \delta(\cdot) \) is the impulse function.

DEFINITION 2:
The mirror function of \( x^*(m_1,m_2) \) is given by

\[ x**(m_1,m_2) = \sum_{m_2} \sum_{m_1} X(N1-1-u,N2-1-v) \delta(u-(N1-1-m1)\Delta u) \delta(v-(N2-1-m2)\Delta v). \quad (3-2) \]
LEMMA 1:
Let $F(w_1,w_2)$ and $F_1(w_1,w_2)$ denote the Fourier transforms of $X^*(m_1,m_2)$ and $X^{**}(m_1,m_2)$ respectively:

$$F(w_1,w_2) = \sum_{m_1=0}^{N_1-1} \sum_{m_2=0}^{N_2-1} X(m_1,m_2) \exp(-jm_1w_1 a u) \exp(-jm_2w_2 a v), \quad (3-3)$$

$$F_1(w_1,w_2) = \sum_{m_1=0}^{N_1-1} \sum_{m_2=0}^{N_2-1} X(N_1-1-m_1, N_2-1-m_2) \exp(-jm_1w_1 a u) \exp(-jm_2w_2 a v). \quad (3-4)$$

Then the relationships between the amplitude and phase spectra of $F_1(w_1,w_2)$ and $F(w_1,w_2)$ are as follows

$$|F(w_1,w_2)| = |F_1(w_1,w_2)|, \quad (3-5)$$

$$\phi(w_1,w_2) = -(N_1-1)w_1 a u - (N_2-1)w_2 a v - \phi_1(w_1,w_2), \quad (3-6)$$

where, $\phi(w_1,w_2)$ is the phase spectrum of $F(w_1,w_2)$ and $\phi_1(w_1,w_2)$ is that of $F_1(w_1,w_2)$.

PROOF:
Eqs.(3-3)(3-4) can be proved directly from the definition of Fourier transform.

Let $y_1 = N_1-1-m_1$ and $y_2 = N_2-1-m_2$, from Eq.(3-4), it follows

$$F_1(w_1,w_2) = \sum_{y_2=0}^{N_2-1} \sum_{y_1=0}^{N_1-1} X(y_1,y_2) \exp(-j(N_1-1-y_1)w_1 a u) \exp(-j(N_2-1-y_2)w_2 a v)$$

$$= \sum_{y_2=0}^{N_2-1} \sum_{y_1=0}^{N_1-1} X(y_1,y_2) \exp(jy_1w_1 a u) \exp(jy_2w_2 a v) \exp(-j(N_1-1)w_1 a u) \exp(-j(N_2-1)w_2 a v), \quad (3-7)$$

Compared Eq.(3-7) with Eq.(3-3), it follows that
F1(w1,w2)=\mathcal{F}(w1,w2)\exp(-j(N1-1)w1u)\exp(-j(N2-1)w2v),

where $\mathcal{F}(w1,w2)$ is the complex conjugate of $\mathcal{F}(w1,w2)$.

Thus, Eqs. (3-5)-(3-6) are proved. \[\text{QED}\]

Represent

\[F1(w1,w2)=R(w1,w2)-jI(w1,w2),\quad (3-8)\]

where

\[R(w1,w2)=\sum_{m1}^{N1-1} \sum_{m2}^{N2-1} X(N1-1-m1, N2-1-m2)\cos(m1w1u+m2w2v),\]

\[I(w1,w2)=\sum_{m1}^{N1-1} \sum_{m2}^{N2-1} X(N1-1-m1, N2-1-m2)\sin(m1w1u+m2w2v).\]

Define a 2-dimensional vector

\[F1(w1,w2)=\begin{bmatrix} R(w1,w2) \\ I(w1,w2) \end{bmatrix} \quad (3-9)\]

Let

\[L1(w1)=\begin{bmatrix} \cos(w1u) & -\sin(w1u) \\ \sin(w1u) & \cos(w1u) \end{bmatrix}, \quad (3-10)\]

\[L2(w2)=\begin{bmatrix} \cos(w2v) & -\sin(w2v) \\ \sin(w2v) & \cos(w2v) \end{bmatrix}. \quad (3-11)\]

It can be shown (Ahmed & Rao, 1975) that $L1(w1)$ and $L2(w2)$ are orthogonal, and hence have the property

\[L1^m(w1)=\begin{bmatrix} \cos(w1u) & -\sin(w1u) \\ \sin(w1u) & \cos(w1u) \end{bmatrix}^m\]
\[
\begin{bmatrix}
\cos(m_1 w_{1A}) & -\sin(m_1 w_{1A}) \\
\sin(m_1 w_{1A}) & \cos(m_1 w_{1A})
\end{bmatrix},
\]

\[
L_2(w_2) = \begin{bmatrix}
\cos(w_{2A}) & -\sin(w_{2A}) \\
\sin(w_{2A}) & \cos(w_{2A})
\end{bmatrix}^{m_2}
\]

\[
= \begin{bmatrix}
\cos(m_2 w_{2A}) & -\sin(m_2 w_{2A}) \\
\sin(m_2 w_{2A}) & \cos(m_2 w_{2A})
\end{bmatrix},
\]

Hence

\[
F_1(w_1, w_2) = \sum_{m_2=0}^{m_2} L_2(w_2) L_1(w_1) e^{X(N_1-1-m_1, N_2-1-m_2)},
\]

where
\[
e = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.
\]

Denote

\[
F_2(w_1, N_2-1-m_2) = \sum_{m_1=0}^{N_1-1} L_1(w_1) e^{X(N_1-1-m_1, N_2-1-m_2)}.
\]

Then, (3-14) can be rewritten as

\[
F_1(w_1, w_2) = \sum_{m_2=0}^{m_2} L_2(w_2) \sum_{m_1=0}^{N_1-1} L_1(w_1) e^{X(N_1-1-m_1, N_2-1-m_2)}
\]

\[
= \sum_{m_2=0}^{m_2} L_2(w_2) F_2(w_1, N_2-1-m_2).
\]

**L\text{EMMA 2}:**

Define the recursive formula

\[
Z_2(w_1, w_2, S_2) = L_2(w_2) Z_2(w_1, w_2, S_2-1) + F_2(w_1, S_2),
\]

\[
Z_2(w_1, w_2, -1) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.
\]
It follows that
\[
Z_2(w_1,w_2,N_2-1) = \sum_{m_2=0}^{N_2-1} L_2(w_2) F_2(w_1,N_2-1-m_2)
\]
\[= F_1(w_1,w_2). \quad (3-18)
\]

**PROOF:** (using mathematical induction)

1. For $S_2=0$, $Z_2(w_1,w_2,0)=F_2(w_1,0)$.

2. Assume, $Z_2(w_1,w_2,k) = \sum_{m_2=0}^{K} L_2(w_2) F_2(w_1,k-m_2)$ when $S_2=k$.

3. Hence, when $S_2=k+1$,

\[
Z_2(w_1,w_2,k+1) = L_2(w_2) Z_2(w_1,w_2,k) + F_2(w_1,k+1)
\]
\[= \sum_{m_2=0}^{K} L_2(w_2) F_2(w_1,k+1-m_2) + F_2(w_1,k+1)
\]
\[= \sum_{d=0}^{K} L_2(w_2) F_2(w_1,k+1-d),
\]
where, $m_2+1=d$.

Therefore, when $S_2=N_2-1$, using (3-16), we have

\[
Z_2(w_1,w_2,N_2-1) = \sum_{m_2=0}^{N_2-1} L_2(w_2) F_2(w_1,N_2-1-m_2)
\]
\[= F_1(w_1,w_2). \quad \text{QED}
\]

**NOTE:**

For every $S_2$, $F_2(w_1,S_2)$ is an one dimensional DFT, which can be computed by one dimensional FFT or RFFT.
LEMMA 3: For one dimensional RFFT, the recursive formula for $F_2(w_1,S_2)$ is

$$Z_1(w_1,hN-1,S_2)=L_1^N(w_1)Z_1(w_1,(h-1)N-1,S_2)$$

$$+LX(hN-1,(h-1)N,S_2), \quad h=1,2,\ldots,H,$$

$$Z_1(w_1,-1,S_2)=[0\ 0]^T, \quad (3-19)$$

where $h$ counts the recursive steps along the $u$ axis, $N$ is the number of data for every recursive step ($HN=N_1$) and

$$L=[L_1^0(w_1)e\ L_1^1(w_1)e\ldots\ L_1^{N-1}(w_1)e], \quad (3-20)$$

$$X(hN-1,(h-1)N,S_2)$$

$$=[X(hN-1,S_2),X(hN-2,S_2)\ldots X((h-1)N,S_2)].^T \quad (3-21)$$

PROOF:

Using the same formula as in lemma 1 for $F_2(w_1,S_2)$, we have

$$Z_1(w_1,S_1,S_2)=L_1(w_1)Z_1(w_1,S_1-1)+eX(S_1,S_2), \quad (3-22)$$

$$Z_1(w_1,-1,S_2)=[0\ 0]_m.$$

Therefore,

$$Z_1(w_1,N_1-1,S_2)=\sum_{m_1=0}^{N_1-1} L_1(w_1)eX(N_1-1-m_1,S_2)=F_2(w_1,S_2).$$

In Eq. (3-22), let $S_1=hN-1$; it can be shown that
\[
F_2(w_1, S_2) = Z_1(w_1, hN-1, S_2) = \sum_{m=0}^{hN-1} L_1(w_1) e^{X(hN-1-m, 1, S_2)}.
\]  

(3-23)

Write Eq. (3-23) in the matrix form as follows:

For \( h = 1 \)

\[
Z_1(w_1, N-1, S_2) = [L_1^0(w_1) e L_1^1(w_1) e ... L_1^{N-1}(w_1)e] X^{(N-1, S_2)} + \]

\[
X^{(N-1, S_2)} + ... + X^{(1, S_2)} + X^{(0, S_2)}
\]

(3-24)

For \( h = 2 \)

\[
Z_1(w_1, 2N-1, S_2) = [L_1^0(w_1) e ... L_1^{N-1}(w_1)e L_1^0(w_1) e ... L_1^{2N-1}(w_1)e] X^{(2N-1, S_2)} + \]

\[
X^{(2N-1, S_2)} + ... + X^{(N, S_2)} + X^{(N-1, S_2)} + ... + X^{(1, S_2)} + X^{(0, S_2)}
\]

(3-25)

Eq. (3-25) can be rearranged as follows:

\[
Z_1(w_1, 2N-1, S_2) = [L_1^0(w_1) e ... L_1^{N-1}(w_1)e] X^{(N, S_2)} + \]

\[
X^{(N, S_2)} + ... + X^{(1, S_2)} + X^{(0, S_2)}
\]

(3-26)

Considering Eqs. (3-20)(3-21)(3-24)(3-26), it follows that
\[ Z_1(w_1, 2N-1, S_2) = L_X(2N-1, N, S_2) + L_1^N(w_1)Z_1(w_1, N-1, S_2). \]  

(3-27)

For \( h=3 \), we have

\[ Z_1(w_1, 3N-1, S_2) = [L_1^0(w_1) e \ldots L_1^N(w_1) e L_1^N(w_1) e \ldots L_1^N(w_1) e] X(2N, S_2) \]

\[ L_1^N(w_1) e \ldots L_1^N(w_1) e X(N, S_2) \]

\[ X(N-1, S_2) \]

\[ X(0, S_2) \]

= \[ L_X(3N-1, 2N, S_2) + L_1^N(w_1) Z_1(w_1, 2N-1, S_2). \]

(3-28)

Also, it can be shown (by using mathematical induction similar to that in the lemma 2) that

\[ Z_1(w_1, hN-1, S_2) = L_1^N(w_1) Z_1(w_1, (h-1)N-1, S_2) \]

+ \[ L_X(hN-1, (h-1)N, S_2), \]

\[ Z_1(w_1, -1, S_2) = [0 0]^T. \]  

(3-29)

QED
For convenience, all formulas necessary in the recursive calculation are listed below.

1) For \( F_2(w_1,S_2) = Z_1(w_1,hN-1,S_2) \),

\[
Z_1(w_1,hN-1,S_2) = L_1^N(w_1)Z_1(w_1,(h-1)N-1,S_2) + L_X(hN-1,(h-1)N,S_2),
\]

\[ h=1,2,3,\ldots,H, \]

\[
Z_1(w_1,-1,S_2) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \text{and} \quad e = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.
\]

2) For \( F_1(w_1,w_2) = Z_2(w_1,w_2,N2-1) \),

\[
Z_2(w_1,w_2,S_2) = L_2(w_2)Z_2(w_1,w_2,S_2-1) + Z_1(w_1,N1-1,S_2),
\]

\[ S_2 = 0,1,2,\ldots,N2-1, \]

\[
Z_2(w_1,w_2,-1) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.
\]

3) \( |F(w_1,w_2)| = |Z_2(w_1,w_2,N2-1)| \),

\[
\Phi(w_1,w_2) = -((N1-1)w_1\Delta u + (N2-1)w_2\Delta v

- \arctg(-I(w_1,w_2)/R(w_1,w_2))).
\]
2. FAST ALGORITHM OF RFFT

Eq. (3-30) is a recursive formula for processing all data segment by segment (N sample data per segment). The inspection of Eq. (3-30) shows that the amount of calculations is mainly determined by the term of \( LX(hN-1, (h-1)N, S2) \). Hence, this term should be computed by revised FFT. The main idea in this section is to make use of the butterfly diagram of general FFT (see Fig. 1). But since the number \( N \) of sample data is not equal to the number \( A1 \) of frequencies (this statement will be discussed below), a revised butterfly diagram is necessary to process these sample data. Next, the revised FFT algorithm is discussed and a butterfly diagram (see Fig. 3) similar to the butterfly diagram of general FFT is obtained.

Let

\[
X(hN-1, (h-1)N, S2) = [Xh(0, S2) Xh(1, S2) \ldots Xh(N-1, S2)]^T,
\]

(3-34)
where, \([Xh(0,S2),Xh(1,S2)...Xh(N-1,S2)]\) denotes the inverse sampled data sequence in the \(h\)-th segment and the \(S2\)-th row, i.e.,

\[
Xh(0,S2)=X(hN-1,S2), \\
Xh(1,S2)=X(hN-2,S2), \\
\vdots \\
Xh(N-1,S2)=X((h-1)N,S2).
\]

Then

\[
LX(hN-1,(h-1)N,S2)=[L1^p(w1)e...L1^{N-1}(w1)e]Xh(0,S2) \\
| \\
|Xh(N-1,S2)]
\]

\[
= \left[ \sum_{p=0}^{N-1} Xh(p,S2) \cos(p\omega 1\Delta u) \right] \\
\sum_{p=0}^{N-1} Xh(p,S2) \sin(p\omega 1\Delta u) \right].
\]

(3-35)

Assume that the bandwidth of the signal is limited. The maximum frequencies along \(w1\) and \(w2\) axes are \(f_{\text{max}1}\) and \(f_{\text{max}2}\) respectively. From sampling theorem, we should have \(\Delta u \leq 1/2f_{\text{max}1}\) and \(\Delta v \leq 1/2f_{\text{max}2}\). Select \(\Delta u=1/2f_{\text{max}1}\), \(\Delta v=1/2f_{\text{max}2}\), and suppose that \(A1\) and \(A2\) are the numbers of frequencies distributed between 0 and \(2f_{\text{max}1}\), and 0 and \(2f_{\text{max}2}\), respectively. Therefore,

\[
w1=k1\Delta w1=k1x2\pi f_{\text{max}1} \quad x2/A1=k1x2\pi /A1\Delta u, \quad (3-36)
\]

\[
w2=k2x2\pi /A2\Delta v. \quad (3-37)
\]

From Eqs. (3-35) (3-36), define
\[ C(k_1) = \sum_{p=0}^{N-1} X_h(p,s_2) \exp(-j2\pi pk_1/A_1), \]
\[ k_1=0,1,2,\ldots,A_1-1. \]  
(3-38)

Eq. (3-38) allows to obtain \( A_1 \) frequencies from \( N \) sampled data. This is one of differences between RFFT and general FFT, since for FFT, the constraint of \( A_1=N \) must be satisfied. The relation between \( A_1 \) and \( N \) and the computational procedure for Eq. (3-38) will be shown in the following example.

**Notation**: Each decimal value of \( p \), \( 0 \leq p \leq N-1 \), is expressed in binary form

\[ p = m_n 2^n + m_{n-1} 2^{n-1} + \ldots + m_1 2^1 + m_0 2^0, \]

where \( m_i = 0,1 \), \( i=0,1\ldots n-1 \), \( n = \log_2 N \).

Similarly each decimal value of \( k_1 \), \( 0 \leq k_1 \leq A_1-1 \), is expressed as

\[ k_1 = k_a 2^a + k_{a-1} 2^{a-1} + \ldots + k_1 2 + k_0 2^0, \]

where \( k_i = 0,1 \), \( i=0,1\ldots a-1 \), \( a = \log_2 A_1 \).

Denote the binary representation of \( X_h(p,s_2) \) by

\[ X_h(p,s_2) = D_h(m_n, m_{n-1} \ldots m_1, m_0, s_2). \]  
(3-39)

Eqs. (3-38) (3-39) lead to the relation

\[ C(k_1) = \sum_{p=0}^{N-1} X_h(p,s_2)W^{kp} \]
\[ = \sum_{m_n} \sum_{m_{n-1}} \ldots \sum_{m_1} D_h(m_n, m_{n-1} \ldots m_1, s_2)W^{k_1(m_n, m_{n-1} \ldots m_1, s_2)} \]
\[ = \sum_{m_n} \sum_{m_{n-1}} \ldots \sum_{m_1} D_h(m_n, m_{n-1} \ldots m_1, s_2)W^{k_1(m_n, 2^{n-1} + \ldots + m_1 2^1 + m_0 2^0)}. \]  
(3-40)

where \( W = \exp(-j2\pi/A_1) \).
Let N=4, A1=8, thus n=2, a=3, W=exp(-j2π/8). For this case

\[ C(k_1) = \sum_{m_0} \sum_{m_i} Dh(m_i, m_o, S2) W^{kl(m; z^2 + m^2)} \]  

(3-41)

Assume,

\[ M_1 = \sum_{m_i} Dh(m_i, m_o, S2) W^{k^2m_i^2} = \sum_{m_i} Dh(m_i, m_o, S2) W^{2m_i(2k_2 + 2k_1 + K_o)} \]

(3-42)

Noting that W=1, we obtain

\[ M_1 = \sum_{m_i} Dh(m_i, m_o, S2) W^{2m_i(2k_2 + K_o)} \]

(3-42)

The summation over \( m_i \) in Eq.(3-42) results in a function of \( k_o, k_i \) and \( m_o \), which is denoted by \( X_1(k_o, k_i, m_o) \), i.e.,

\[ M_1 = X_1(k_o, k_i, m_o) \]

(3-43)

Replacing Eq.(3-43) into Eq.(3-41), we have

\[ C(k_1) = \sum_{m_o} X_1(k_o, k_i, m_o) W^{klm_o} = \sum_{m_o} X_1(k_o, k_i, m_o) W^{m_o(2k_2 + 2k_1 + K_o)} \]

Similarly

\[ C(k_1) = X_2(k_o, k_i, k_2) = M_0 \]

(3-44)

With respect to \( X_1(k_o, k_i, m_o) \) (i.e., \( M_1 \)),

\[ X_1(k_o, k_i, m_o) = Dh(0, m_o, S2) + Dh(1, m_o, S2) W^{m_o(2k_2 + 2k_1 + K_o)} \]

(3-45)

**Case 1.** \( k_i, k_o = 0, 0 \),

\[ X_1(0, 0, 0) = Dh(0, 0, S2) + Dh(1, 0, S2) W, \]

\[ X_1(0, 0, 1) = Dh(0, 1, S2) + Dh(1, 1, S2) W^o. \]

**Case 2.** \( k_i, k_o = 0, 1 \),
\[
X_{1}(1,0,0) = Dh(0,0,S2) + Dh(1,0,S2)W^2, \\
X_{1}(1,0,1) = Dh(0,1,S2) + Dh(1,1,S2)W^2.
\]

**Case 3.** \(k, k_0 = 1, 0\),
\[
X_{1}(0,1,0) = Dh(0,0,S2) + Dh(1,0,S2)W^4, \\
X_{1}(0,1,1) = Dh(0,1,S2) + Dh(1,1,S2)W^4.
\]

**Case 4.** \(k, k_0 = 1, 1\),
\[
X_{1}(1,1,0) = Dh(0,0,S2) + Dh(1,0,S2)W^6, \\
X_{1}(1,1,1) = Dh(0,1,S2) + Dh(1,1,S2)W^6.
\]

Similarly, with respect to \(X_2(k, k_1, k_2)\) (i.e. \(M_0\)), it can be obtained
\[
X_2(k, k_1, k_2) = X_1(k_0, k_1, 0) + X_1(k_0, k_1, 1)W^{4k_2 + 2k_1 + k_0}. \tag{3-46}
\]

**Case 1.** \(k_2 k, k_0 = 0, 0, 0\),
\[
X_2(0,0,0) = X_1(0,0,0) + X_1(0,0,1)W^6.
\]

**Case 2.** \(k_2 k, k_0 = 0, 0, 1\),
\[
X_2(1,0,0) = X_1(1,0,0) + X_1(1,0,1)W^4.
\]

**Case 3.** \(k_2 k, k_0 = 0, 1, 0\),
\[
X_2(0,1,0) = X_1(0,1,0) + X_1(0,1,1)W^2.
\]

**Case 4.** \(k_2 k, k_0 = 0, 1, 1\),
\[
X_2(1,1,0) = X_1(1,1,0) + X_1(1,1,1)W^6.
\]

**Case 5.** \(k_2 k, k_0 = 1, 0, 0\),
\[
X_2(0,0,1) = X_1(0,0,0) + X_1(0,0,1)W^4.
\]
Case 6. \( k_2 k_0 = 101 \),
\[ X_2(1,0,1) = X_1(1,0,0) + X_1(1,0,1)W^5 . \]

Case 7. \( k_2 k_0 = 110 \),
\[ X_2(0,1,1) = X_1(0,1,0) + X_1(0,1,1)W^6 . \]

Case 8. \( k_2 k_0 = 111 \),
\[ X_2(1,1,1) = X_1(1,1,0) + X_1(1,1,1)W^7 . \]

This sequence of arithmetic operations is shown in Fig. 3.

![8-point Butterfly Diagram of Revised FFT](image)

8-point Butterfly Diagram of Revised FFT

Fig. 3

Here, the sample data \( D_h(m_1, m_2, S_2) \) expressed in binary form are replaced by the data \( X_h(p, S_2) \) expressed in decimal form according to Eq. (3-39). From the complex conjugate property, we have
C(A1/2+L) = C*(A1/2-L), \quad L=1,2,\ldots\ldots A1/2-1,

where, C*( ) is the complex conjugate of C( ).

The C(k1), k1=0,1,2,3,4, is calculated and others can be obtained by the conjugate property of DFT.

This completes the construction of revised FFT algorithm. Representing C(k1) as real and imaginary parts

\[ C(k1) = R1(k1) - jI1(k1), \quad (3-47) \]

and using Eq. (3-38), we have

\[ R1(k1) = \sum_{p=0}^{N-1} Xh(p,S2) \cos(2\pi pk1/A1), \quad (3-48) \]
\[ I1(k1) = \sum_{p=0}^{N-1} Xh(p,S2) \sin(2\pi pk1/A1). \quad (3-49) \]

Considering Eqs. (3-36)(3-37) and substituting Eqs. (3-48)(3-49) into Eq. (3-35), \( LX(hN-1,(h-1)N,S2) \) can be obtained. By comparing Fig.3 with Fig.1, it is seen that the butterfly diagram of the revised FFT is similar to that of general FFT. The difference between them is only in the starting of iteration. Because of \( a > n \) (see page 30), the first iteration for Eq. (3-38) agrees with the \((a-n+1)\)-th iteration of general FFT. The sample data are used repeatedly, and the number of repetitions is \( r=A1/N \).
CHAPTER IV

EXPERIMENTAL RESULTS

In order to illustrate the possible applications of 2-D RFFT algorithm, three experiments are presented in this chapter. In the first experiment, a 2-D exponential function is transformed by RFFT and the results are compared with DFT, FFT, and ideal spectrum. In the second experiment, a more complicated function is shown, and in the third one, a signal detection using the RFFT is simulated.

The following is a calculational diagram of the algorithm: when the new data appear, one dimensional RFFT or revised FFT is first used to transform these data to $F_2(w_1,S_2)$, and then $F_1(w_1,w_2)$ is computed by the recursive formula; this process continues until all data are used.
N new data in

\[ Z_1(w_1, hN-1, S_2) = L_1''(w_1)Z_1(w_1, (h-1)N-1, S_2) + \]
\[ + L_X(hN-1, (h-1)N, S_2) \]
\[ Z_1(w_1, -1, S_2) = [0], \quad e = [0], \quad h = 1, 2, 3, \ldots, H \]

\[ Z_2(w_1, w_2, S_2) = L_2(w_2)Z_2(w_1, w_2, S_2-1) + \]
\[ + Z_1(w_1, N1-1, S_2) \]
\[ Z_2(w_1, w_2, -1) = [0], \quad S_2 = 0, 1, \ldots, N2-1 \]

Diagram of RFFT Calculation

\[ |F(w_1, w_2)| = |Z_2(w_1, w_2, N2-1)| \]
\[ \Phi(w_1, w_2) = -((N1-1)w_1u + (N2-1)w_2v) \]
\[ - \text{arctg}(-I(w_1, w_2)/R(w_1, w_2)) \]

Fig. 4
Experiment 1:

For a two dimensional signal:

\[ X(u,v) = \exp(-u-v) \quad u \geq 0, \ v \geq 0, \quad \text{(4-1)} \]

its Fourier transform, i.e. the ideal spectrum, is

\[ F(w_1,w_2) = \frac{1}{jw_1+1} \frac{1}{jw_2+1}, \quad \text{(4-2)} \]

and the amplitude and phase spectra are

\[ |F(w_1,w_2)| = \frac{1}{\sqrt{(w_1^2+1)(w_2^2+1))}}, \quad \text{(4-3)} \]

\[ \phi(w_1,w_2) = -[\arctg(w_1)+\arctg(w_2)]. \quad \text{(4-4)} \]

Table 1 shows the ideal spectrum. Fig.5, Fig.6, Table 3 and Table 5 to 7 are the spectra by using 2-D RFFT. From these experimental results, it is noticed that as the data increase, the spectra tend to the ideal spectrum monotonically.

Comparing Table 2, which represents the spectrum obtained by DFT, with Table 3, it can be noticed that the results of RFFT and DFT are identical. RFFT has the same advantage as DFT that the number of sample data may not be equal to the number of frequencies. The transform speed of RFFT is faster than that of DFT because the fast algorithm is used. Table 8 presents the spectrum obtained
by RFFT with half sample interval of Table 2. Compared Table 8 with Table 3, it is seen that reducing sample interval makes the spectrum closer to the ideal spectrum, because reducing sample interval implies reducing the overlap of the spectrum.

Comparing Table 4, which represents the FFT algorithm results, with Table 6, it is seen that RFFT and FFT yield the same result when the number of sample data equals that of frequencies.

**Experiment 2:**

In this experiment, a complicated two-dimensional signal to be transformed comes from the model of an ideal low-pass filter and is revised here for the purpose of practical realization.

**Definition:**

\[
X(u,v) = \begin{cases} 
\exp\left(-0.1v+u^2/4v\right)/v^{1/2} & u \geq 1, v > 0 \\
0 & u < 1, v < 0 
\end{cases} 
\]

(4-5)

The Fourier Transform of \( X(u,v) \) is

\[
F\{X(u,v)\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp\left(-0.1v+u^2/4v\right)/v^{1/2} \exp\left(-jw2v+jw1u\right) dv du
\]

\[
= \left[ \pi/(0.1+jw2) \right]^{1/2} \exp[-u((0.1+jw2)^{1/2}+jw1)] du
\]
\[ \pi \exp(-{(jw_1 + (0.1 + jw_2)^{1/2})}/(0.1 + jw_2)^{1/2}; jw_1 + (0.1 + jw_2)^{1/2}). \]

\[ (4-6) \]

The amplitude spectrum of \( F\{X(u,v)\} \) can be obtained as
\[
|F\{X(u,v)\}| = \pi \exp(-(0.1^2 + w_2^2)^{1/4} \cos(\theta/2))/(\{w_1 + (0.1^2 + w_2^2)^{1/2}
\]
\[
(\sin(\theta/2))^2 + (0.1^2 + w_2^2)^{1/2} \cos^2(\theta/2)) \} (0.1^2 + w_2^2)^{1/2},
\]
\[ (4-7) \]

where, \( \theta = \arctg(10w_2) \).

The Fig. 7 to Fig. 13 show the convergence of the spectra to the ideal spectrum by using 2-D RFFT. Fig. 13 shows a good agreement between the ideal spectrum and the result obtained from 2-D RFFT.

Experiment 3:

This is a simulation of signal detection by using RFFT. The signal \( g(u,v) \) is mixed with additive noise \( n(u,v) \) and the time of the signal appearance \((a, b)\) is unknown. The input signal is
\[
X(u,v)=g(u-a,v-b)U(u-a,v-b)+n(u,v)U(u,v), \quad a\geq 0, \ b\geq 0,
\]
\[ (4-8) \]

where \( U(u,v) \) is a 2-D unit step function.

Assume that the noise \( n(u,v) \) is Gaussian, white noise with zero mean and unit variance and has stationary and ergodic
properties. Assume also that $g(u,v)$ and $n(u,v)$ are uncorrelated. Resulting from these assumptions, the autocorrelation function of $X(u,v)$ is

$$Rx(z_1,z_2) = R_g(z_1,z_2) + R_n(z_1,z_2). \quad (4-9)$$

And the power spectrum of $X(u,v)$ is

$$S_x(w_1,w_2) = S_g(w_1,w_2) + S_n(w_1,w_2), \quad (4-10)$$

where

$$Rx(z_1,z_2) = \lim_{T_1 \to \infty, T_2 \to \infty} \frac{1}{T_1 T_2} \int_{-T_1}^{T_1} \int_{-T_2}^{T_2} X(u,v)X(u-z_1,v-z_2) \, du \, dv,$$

$$R_g(z_1,z_2) = \lim_{T_1 \to \infty, T_2 \to \infty} \frac{1}{T_1 T_2} \int_{-T_1}^{T_1} \int_{-T_2}^{T_2} g(u,v)g(u-z_1,v-z_2) \, du \, dv,$$

$$R_n(z_1,z_2) = \lim_{T_1 \to \infty, T_2 \to \infty} \frac{1}{T_1 T_2} \int_{-T_1}^{T_1} \int_{-T_2}^{T_2} n(u,v)n(u-z_1,v-z_2) \, du \, dv,$$

$$S_x(w_1,w_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} Rx(z_1,z_2) \exp(-jw_1z_1+jw_2z_2) \, dz_1 \, dz_2,$$

$$S_g(w_1,w_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} R_g(z_1,z_2) \exp(-jw_1z_1+jw_2z_2) \, dz_1 \, dz_2,$$

$$S_n(w_1,w_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} R_n(z_1,z_2) \exp(-jw_1z_1+jw_2z_2) \, dz_1 \, dz_2.$$

If no signal is present, i.e., $g(u,v) = 0$, then

$$S_x(w_1,w_2) = S_n(w_1,w_2) = 1.$$

The key point here is to obtain $S_x(w_1,w_2)$ from the random
signal $X(u,v)$ directly, because the calculation of correlation $R_x(z_1,z_2)$ requires all sample data and therefore does not allow the recursive method to be used for on-line applications.

Assuming that $X(u,v)$ is ergodic, the following formula can be derived

$$S_x(w_1,w_2)=\lim_{T_1,T_2}\int \int X(u,v)\exp((-jw_1u+jw_2v))du dv/T_1T_2.$$  \hspace{1cm} (4-11)

Because the detection of sinusoidal/narrow-band signals has received a great deal of attention (Kumaresan & Tufts, 1983) (Glover, 1977), the sinusoidal signal is selected as a signal to be detected in this experiment, i.e.,

$$g(u,v)=\sin(\pi u-\pi v).$$  \hspace{1cm} (4-12)

The discrete form of Eq.(4-11) is

$$S_x^*(k_1,k_2)=\sum_{m_1=0}^{N_1-1}\sum_{m_2=0}^{N_2-1}X(m_1,m_2)\exp(-2k_1m_1A_1+2k_2m_2A_2)/T_1T_2.$$  \hspace{1cm} (4-13)

Fig.14 to Fig.17 illustrate the experiment results. At the beginning, the result is only the noise power spectrum because the collected data does not include the
signal. When the signal appears, the spectrum is changed and the signal is detected better and better with increase of recursive steps (Fig.14 to Fig.17). From the experimental results it is indicated that, in the presence of the additive white noise, RFFT provides an efficient signal detection and accurate estimate of the signal frequency despite a small signal-to-noise ratio (see Fig.18).
CHAPTER V

CONCLUSION AND POSSIBLE APPLICATIONS

The algorithm developed in this thesis provides a recursive fast Fourier transform method, by which the \((A1 \times A2)\) values of frequencies are computed recursively from the \((N1 \times N2)\) sample data sequence. Because of no constraints on the length of signal sequence and recursive steps, the results of transform can approach monotonically the real spectrum as the data sequence-length increases. RFFT is a recursive version of FFT, and theoretically its output is equal to that of general FFT for the same signal sequence. Hence, the convergence of RFFT to the real spectrum can be discussed similarly to that of FFT.

Studying the formulas in Chapter III, we notice that the amount of computational operations of RFFT is larger than that of FFT when the number of sample data is the same as that of frequencies. But from the viewpoint of on-line applications, the transform speed of RFFT might be not lower than that of FFT because part or most computational operations may be finished during the sample
intervals of one segment. This is true if the computer efficiency is relatively high when compared with the duration of one data segment.

The above feature is very suitable for microcomputer or microprocessor implementations. The processes of sampling and calculating may be carried out by several microcomputers or microprocessors separately. This may fully utilize the computer time and speed up the transformation even further, making it very useful for on-line spectral analysis and identification of two dimensional systems.

RFFT may also have other applications. In many cases, the analog-digital transform is necessary. For example, digital Butterworth filter, digital Chebyshev filter (Rader/Gold, 1967) and Elliptic filter (Gold/Rader, 1969) (Guillemin, 1957) (Storer, 1957) all present the design methods for analog-digital transform according to the desired frequency response specifications. Steiglitz (Steiglitz, 1970) (Rabiner/Steiglitz, 1970) has proposed an IIR (infinite impulse response) design procedure based on minimization of mean-square error in frequency domain. All of these methods require a given frequency response as design bases. Thus, if the desired frequency specification comes from a real analog system, all of the methods require the solution to analog-digital transform
problem. With RFFT, all the above procedures might be realized on-line through the identifications of those unknown analog systems in frequency domain. For one dimensional system, the paper (Zhu & Wang, 1986) gives an application of such concept to system identification problem.

Another application of RFFT is to solve the problem of processing a signal sequence which otherwise requires a large computer memory. Especially, in 2-D applications, the large-size data arrays are commonly encountered. Although the signal sequence may be very long, the number of frequencies does not have to be too large for the computer memory to accept. This is because the frequencies in the question are usually distributed in a certain range only. For such a problem, it is difficult to use general FFT (requiring the same number of frequencies and sampling data) to process a large amount of sample data. While selecting the number of data that the computer could accept, the real spectrum will be distorted. RFFT can solve this problem. By selecting the necessary number of frequencies, the desired spectrum could be obtained as the data are processed step by step, regardless of the amount of samples.

Presented here, a "signal detection" experiment (Experiment 3) has shown another advantage of RFFT. Since
the time when the signal appears is unknown, it is necessary to sample and detect the signal continuously, which results in an increasing-size 2-D datum array. If the signal appears after a long delay, then, the amount of sample data will become very large, most of them being the noise. This "useless" information will be transformed, and the number of frequencies produced by FFT will increase greatly because of the large amount of sample data. This may set a big burden on the computers. On the other hand, the signal detected by RFFT can produce the spectrum whose frequency number is usually much smaller than that of sample data in practical applications. Hence, the demand for the computer memory and calculations is relaxed. Moreover, because of the recursive property of RFFT, it is possible to stop the transform as soon as the signal is detected.

In the summary, the recursive property, speed and flexibility of RFFT algorithm make it attractive for the applications in a variety of signal processing and analysis problems.
THE EXPLANATION OF FIGURES AND TABLES

N1: the number of data along horizontal direction.

N: the number of data (per segment) along horizontal direction.

H: the recursive times (for segments) along horizontal direction.

N2: the number of data along vertical direction.

Δu: the sample interval along horizontal direction.

Δv: the sample interval along vertical direction.

A1: the number of frequencies along horizontal direction.

A2: the number of frequencies along vertical direction.
EXP. 1: $X(u,v) = \exp(-(u+v)) \quad u \geq 0, \ v \geq 0$

$N_1 = 128$ (H=4, N=32)
$\Delta u = 0.01$
$A_1 = 128$

$N_2 = 128$
$\Delta v = 0.01$
$A_2 = 128$

Fig. 5
EXP. 1: $X(u,v) = \exp(-(u+v))$  \hspace{1cm} u \geq 0, \hspace{0.1cm} v \geq 0

N1 = 256 (H = 8, N = 32)  \hspace{1cm} N2 = 250
\Delta u = 0.01  \hspace{1cm} \Delta v = 0.01
A1 = 128  \hspace{1cm} A2 = 128

Fig. 6
EXP. 2: $X(u,v) = \begin{cases} \exp\left(-0.1v + u^2/4v\right)/v^{1/2} & u>1, v>0 \\ 0 & u<1, v<0 \end{cases}$

$N_1 = 64$ (H=1, N=64) $\Delta u = 0.1$ $A_1 = 128$

$N_2 = 50$ $\Delta v = 0.1$ $A_2 = 128$

Fig. 7

EXP. 2: $X(u,v) = \begin{cases} \exp\left(-0.1v + u^2/4v\right)/v^{1/2} & u>1, v>0 \\ 0 & u<1, v<0 \end{cases}$

$N_1 = 64$ (H=1, N=64) $\Delta u = 0.1$ $A_1 = 128$

$N_2 = 100$ $\Delta v = 0.1$ $A_2 = 128$

Fig. 8
EXP. 2: \( X(u, v) = \begin{cases} \exp\left(-0.1v + u^2/4v\right)/v^{1/2} & u \geq 1, v > 0 \\ 0 & u < 1, v < 0 \end{cases} \)

\(N_1 = 128\) (\(H = 1, N = 128\))
\(\Delta u = 0.1\)
\(A_1 = 128\)

\(N_2 = 128\)
\(\Delta v = 0.1\)
\(A_2 = 128\)

Fig. 9

EXP. 2: \( X(u, v) = \begin{cases} \exp\left(-0.1v + u^2/4v\right)/v^{1/2} & u \geq 1, v > 0 \\ 0 & u < 1, v < 0 \end{cases} \)

\(N_1 = 128\) (\(H = 2, N = 64\))
\(\Delta u = 0.1\)
\(A_1 = 128\)

\(N_2 = 128\)
\(\Delta v = 0.1\)
\(A_2 = 128\)

Fig. 10
EXP. 2: \( X(u,v) = \begin{cases} \exp(-(0.1v+u^2/4v))/v^{1/2} & u \geq 1, v \geq 0 \\ 0 & u < 1, v < 0 \end{cases} \)

\begin{align*}
N_1 &= 192 \ (H=3, \ N=64) \\
N_2 &= 200 \\
\Delta u &= 0.1 \\
\Delta v &= 0.1 \\
A_1 &= 128 \\
A_2 &= 128
\end{align*}

Fig. 11

EXP. 2: \( X(u,v) = \begin{cases} \exp(-(0.1v+u^2/4v))/v^{1/2} & u \geq 1, v \geq 0 \\ 0 & u < 1, v < 0 \end{cases} \)

\begin{align*}
N_1 &= 256 \ (H=2, \ N=128) \\
N_2 &= 200 \\
\Delta u &= 0.1 \\
\Delta v &= 0.1 \\
A_1 &= 128 \\
A_2 &= 128
\end{align*}

Fig. 12
IDEAL SPECTRUM

EXP. 2: \( X(u, v) = \begin{cases} \exp\left(-\left(0.1v + u^2/4v\right)/v^{1/2}\right) & u \geq 1, v \geq 0 \\ 0 & u < 1, v < 0 \end{cases} \)

\(N_1 = 384\) (\(H = 3, N = 128\))  \(N_2 = 350\)
\(\Delta u = 0.1\)  \(\Delta v = 0.1\)
\(A_1 = 128\)  \(A_2 = 128\)

Fig. 13
EXP. 3: Spectrum of Two Dimensional White Gaussian Noise n(u,v)

Fig. 14

EXP. 3: $X(u,v) = \sin(\pi(u-4)-\pi(v-4))U(u-4,v-4)+n(u,v)U(u,v)$

$N_1=112$ (H=7, N=16)  
$\Delta u=0.125$  
$A_1=64$

$N_2=112$  
$\Delta v=0.125$  
$A_2=64$

Fig. 15
EXP. 3: \( X(u,v) = \sin(\pi(u-4) - \pi(v-4))U(u-4,v-4) + n(u,v)U(u,v) \)

- \( N1 = 128 \) (\( H=4 \), \( N=32 \))
- \( \Delta u = 0.125 \)
- \( A1 = 64 \)

- \( N2 = 128 \)
- \( \Delta v = 0.125 \)
- \( A2 = 64 \)

Fig. 16

EXP. 3: \( X(u,v) = \sin(\pi(u-4) - \pi(v-4))U(u-4,v-4) + n(u,v)U(u,v) \)

- \( N1 = 160 \) (\( H=5 \), \( N=32 \))
- \( \Delta u = 0.125 \)
- \( A1 = 64 \)

- \( N2 = 150 \)
- \( \Delta v = 0.125 \)
- \( A2 = 64 \)

Fig. 17
EXP. 3: $X(u,v) = 0.1 \sin(\pi(u-4) - \pi(v-4))U(u-4,v-4) + n(u,v)U(u,v)$

$N1 = 640 \ (H=10, N=64)$
$N2 = 640$
$\Delta u = 0.125$
$\Delta v = 0.125$
$A1 = 64$
$A2 = 64$

Fig. 18
**SPECTRUM ANALYSIS WITH TRANSFER FUNCTION**

Function: \( f(x, y) = \exp(-(x \cdot c1 + y \cdot c2)) \)
Transfer Function: \( F(S_1, S_2) = \frac{1}{S_1 + C_1} \cdot \frac{1}{S_2 + C_2} \)

Factor: 
- \( C_1 = 1.0000 \)
- \( C_2 = 1.0000 \)

**RE = REAL \( F(W_1, W_2) \), IM = IMAG \( F(W_1, W_2) \), MOD = MODEL \( F(W_1, W_2) \), ANL = ANGLE \( F(W_1, W_2) \)**

<table>
<thead>
<tr>
<th>( W_1 )</th>
<th>( W_2 )</th>
<th>RE</th>
<th>IM</th>
<th>MOD</th>
<th>ANL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1.0000</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0.0609</td>
<td>-0.2391</td>
<td>-0.2467</td>
<td>-0.2467</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0.0153</td>
<td>-0.1253</td>
<td>-0.1253</td>
<td>-0.1253</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0.0072</td>
<td>-0.0428</td>
<td>-0.0428</td>
<td>-0.0428</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0.0040</td>
<td>-0.0634</td>
<td>-0.0634</td>
<td>-0.0634</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0.0025</td>
<td>-0.0511</td>
<td>-0.0511</td>
<td>-0.0511</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0.0018</td>
<td>-0.0385</td>
<td>-0.0385</td>
<td>-0.0385</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0.0013</td>
<td>-0.0263</td>
<td>-0.0263</td>
<td>-0.0263</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0.0010</td>
<td>-0.0179</td>
<td>-0.0179</td>
<td>-0.0179</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0.0009</td>
<td>-0.0162</td>
<td>-0.0162</td>
<td>-0.0162</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0.0009</td>
<td>-0.0145</td>
<td>-0.0145</td>
<td>-0.0145</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0.0008</td>
<td>-0.0121</td>
<td>-0.0121</td>
<td>-0.0121</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0.0007</td>
<td>-0.0098</td>
<td>-0.0098</td>
<td>-0.0098</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0.0006</td>
<td>-0.0080</td>
<td>-0.0080</td>
<td>-0.0080</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0.0005</td>
<td>-0.0064</td>
<td>-0.0064</td>
<td>-0.0064</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0.0004</td>
<td>-0.0050</td>
<td>-0.0050</td>
<td>-0.0050</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>0.0004</td>
<td>-0.0036</td>
<td>-0.0036</td>
<td>-0.0036</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0.0003</td>
<td>-0.0023</td>
<td>-0.0023</td>
<td>-0.0023</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>0.0003</td>
<td>-0.0013</td>
<td>-0.0013</td>
<td>-0.0013</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0.0002</td>
<td>-0.0003</td>
<td>-0.0003</td>
<td>-0.0003</td>
</tr>
</tbody>
</table>

**TABLE 1** Spectrum Analysis with Transfer Function: \( F(W_1, W_2) = \frac{1}{(jW_1 + 1)(jW_2 + 1)} \)
**SPECTRUM ANALYSIS WITH DFT**

**INPUT FUNCTION:** \( f(x,y) = \exp(-x^2 - y^2) \)

**FACTOR:** \( c_1 = 1.0000, c_2 = 1.0000 \)

**NUMBER OF INPUT:** \( N_1 = 28, N_2 = 30 \)

**NUMBER OF SPECTRUM:** \( N_f = 16, N_y = 16 \)

**INPUT SAMPLE STEP:** \( T = 0.1000 \)

---

<table>
<thead>
<tr>
<th>U1</th>
<th>0</th>
<th>W1 = 0</th>
<th>W2 = 0</th>
<th>RE = REAL(( F(U1,W1) ))</th>
<th>IM = IMAG(( F(U1,W1) ))</th>
<th>MOD = MODEL(( F(U1,W1) ))</th>
<th>ANL = ANGLE(( F(U1,W1) ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1</td>
<td>1</td>
<td>W1 = 0</td>
<td>W2 = 0</td>
<td>RE = 0.065468</td>
<td>IM = 0.000000</td>
<td>MOD = 0.065468</td>
<td>ANL = 0.000000</td>
</tr>
<tr>
<td>U1</td>
<td>2</td>
<td>W1 = 0</td>
<td>W2 = 0</td>
<td>RE = 0.0972049</td>
<td>IM = -0.044998</td>
<td>MOD = 0.144264</td>
<td>ANL = 0.094265</td>
</tr>
<tr>
<td>U1</td>
<td>3</td>
<td>W1 = 0</td>
<td>W2 = 0</td>
<td>RE = 0.070770</td>
<td>IM = -0.125713</td>
<td>MOD = 0.144264</td>
<td>ANL = 0.094265</td>
</tr>
<tr>
<td>U1</td>
<td>4</td>
<td>W1 = 0</td>
<td>W2 = 0</td>
<td>RE = 0.062469</td>
<td>IM = -0.070593</td>
<td>MOD = 0.144264</td>
<td>ANL = 0.094265</td>
</tr>
<tr>
<td>U1</td>
<td>5</td>
<td>W1 = 0</td>
<td>W2 = 0</td>
<td>RE = 0.051563</td>
<td>IM = -0.046556</td>
<td>MOD = 0.144264</td>
<td>ANL = 0.094265</td>
</tr>
<tr>
<td>U1</td>
<td>6</td>
<td>W1 = 0</td>
<td>W2 = 0</td>
<td>RE = 0.051508</td>
<td>IM = -0.036494</td>
<td>MOD = 0.144264</td>
<td>ANL = 0.094265</td>
</tr>
<tr>
<td>U1</td>
<td>7</td>
<td>W1 = 0</td>
<td>W2 = 0</td>
<td>RE = 0.056060</td>
<td>IM = -0.021673</td>
<td>MOD = 0.144264</td>
<td>ANL = 0.094265</td>
</tr>
<tr>
<td>U1</td>
<td>8</td>
<td>W1 = 0</td>
<td>W2 = 0</td>
<td>RE = 0.053121</td>
<td>IM = -0.006712</td>
<td>MOD = 0.144264</td>
<td>ANL = 0.094265</td>
</tr>
<tr>
<td>U1</td>
<td>9</td>
<td>W1 = 0</td>
<td>W2 = 0</td>
<td>RE = 0.049232</td>
<td>IM = -0.000000</td>
<td>MOD = 0.144264</td>
<td>ANL = 0.094265</td>
</tr>
<tr>
<td>U1</td>
<td>10</td>
<td>W1 = 0</td>
<td>W2 = 0</td>
<td>RE = 0.053121</td>
<td>IM = -0.006711</td>
<td>MOD = 0.144264</td>
<td>ANL = 0.094265</td>
</tr>
<tr>
<td>U1</td>
<td>11</td>
<td>W1 = 0</td>
<td>W2 = 0</td>
<td>RE = 0.056060</td>
<td>IM = -0.021673</td>
<td>MOD = 0.144264</td>
<td>ANL = 0.094265</td>
</tr>
<tr>
<td>U1</td>
<td>12</td>
<td>W1 = 0</td>
<td>W2 = 0</td>
<td>RE = 0.051508</td>
<td>IM = -0.036494</td>
<td>MOD = 0.144264</td>
<td>ANL = 0.094265</td>
</tr>
<tr>
<td>U1</td>
<td>13</td>
<td>W1 = 0</td>
<td>W2 = 0</td>
<td>RE = 0.051563</td>
<td>IM = -0.046556</td>
<td>MOD = 0.144264</td>
<td>ANL = 0.094265</td>
</tr>
<tr>
<td>U1</td>
<td>14</td>
<td>W1 = 0</td>
<td>W2 = 0</td>
<td>RE = 0.062469</td>
<td>IM = -0.070593</td>
<td>MOD = 0.144264</td>
<td>ANL = 0.094265</td>
</tr>
<tr>
<td>U1</td>
<td>15</td>
<td>W1 = 0</td>
<td>W2 = 0</td>
<td>RE = 0.070770</td>
<td>IM = -0.125713</td>
<td>MOD = 0.144264</td>
<td>ANL = 0.094265</td>
</tr>
<tr>
<td>U1</td>
<td>16</td>
<td>W1 = 0</td>
<td>W2 = 0</td>
<td>RE = 0.0972249</td>
<td>IM = -0.228467</td>
<td>MOD = 0.248677</td>
<td>ANL = 0.2293910</td>
</tr>
<tr>
<td>U1</td>
<td>17</td>
<td>W1 = 0</td>
<td>W2 = 0</td>
<td>RE = -0.046482</td>
<td>IM = -0.046030</td>
<td>MOD = 0.065803</td>
<td>ANL = 0.225.127620</td>
</tr>
<tr>
<td>U1</td>
<td>18</td>
<td>W1 = 0</td>
<td>W2 = 0</td>
<td>RE = -0.020934</td>
<td>IM = -0.020804</td>
<td>MOD = 0.020804</td>
<td>ANL = 0.223.636200</td>
</tr>
<tr>
<td>U1</td>
<td>19</td>
<td>W1 = 0</td>
<td>W2 = 0</td>
<td>RE = -0.010141</td>
<td>IM = -0.010141</td>
<td>MOD = 0.010141</td>
<td>ANL = 0.224.746800</td>
</tr>
<tr>
<td>U1</td>
<td>20</td>
<td>W1 = 0</td>
<td>W2 = 0</td>
<td>RE = -0.005679</td>
<td>IM = -0.005679</td>
<td>MOD = 0.005679</td>
<td>ANL = 0.225.118930</td>
</tr>
<tr>
<td>U1</td>
<td>21</td>
<td>W1 = 0</td>
<td>W2 = 0</td>
<td>RE = -0.00320</td>
<td>IM = -0.00320</td>
<td>MOD = 0.00320</td>
<td>ANL = 0.227.940920</td>
</tr>
</tbody>
</table>

**TABLE 2** Spectrum Analysis with DFT (\( N_1 = 28, N_2 = 30 \))
### TABLE 3 Spectrum Analysis with RFFT (N1=28, N2=30)

<table>
<thead>
<tr>
<th>SPECTRUM (W1,W2)</th>
<th>RE = REAL(F(W1,W2))</th>
<th>IM = IMAG(F(W1,W2))</th>
<th>MOD = MODEL(F(W1,W2))</th>
<th>ANL = ANGLE(F(W1,W2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>0</td>
<td>W2</td>
<td>0</td>
<td>RE</td>
</tr>
<tr>
<td>W1</td>
<td>1</td>
<td>W2</td>
<td>0</td>
<td>RE</td>
</tr>
<tr>
<td>W1</td>
<td>2</td>
<td>W2</td>
<td>0</td>
<td>RE</td>
</tr>
<tr>
<td>W1</td>
<td>3</td>
<td>W2</td>
<td>0</td>
<td>RE</td>
</tr>
<tr>
<td>W1</td>
<td>4</td>
<td>W2</td>
<td>0</td>
<td>RE</td>
</tr>
<tr>
<td>W1</td>
<td>5</td>
<td>W2</td>
<td>0</td>
<td>RE</td>
</tr>
<tr>
<td>W1</td>
<td>6</td>
<td>W2</td>
<td>0</td>
<td>RE</td>
</tr>
<tr>
<td>W1</td>
<td>7</td>
<td>W2</td>
<td>0</td>
<td>RE</td>
</tr>
<tr>
<td>W1</td>
<td>8</td>
<td>W2</td>
<td>0</td>
<td>RE</td>
</tr>
<tr>
<td>W1</td>
<td>9</td>
<td>W2</td>
<td>0</td>
<td>RE</td>
</tr>
<tr>
<td>W1</td>
<td>10</td>
<td>W2</td>
<td>0</td>
<td>RE</td>
</tr>
<tr>
<td>W1</td>
<td>11</td>
<td>W2</td>
<td>0</td>
<td>RE</td>
</tr>
<tr>
<td>W1</td>
<td>12</td>
<td>W2</td>
<td>0</td>
<td>RE</td>
</tr>
<tr>
<td>W1</td>
<td>13</td>
<td>W2</td>
<td>0</td>
<td>RE</td>
</tr>
<tr>
<td>W1</td>
<td>14</td>
<td>W2</td>
<td>0</td>
<td>RE</td>
</tr>
<tr>
<td>W1</td>
<td>15</td>
<td>W2</td>
<td>0</td>
<td>RE</td>
</tr>
<tr>
<td>W1</td>
<td>16</td>
<td>W2</td>
<td>0</td>
<td>RE</td>
</tr>
<tr>
<td>W1</td>
<td>17</td>
<td>W2</td>
<td>0</td>
<td>RE</td>
</tr>
<tr>
<td>W1</td>
<td>18</td>
<td>W2</td>
<td>0</td>
<td>RE</td>
</tr>
<tr>
<td>W1</td>
<td>19</td>
<td>W2</td>
<td>0</td>
<td>RE</td>
</tr>
<tr>
<td>W1</td>
<td>20</td>
<td>W2</td>
<td>0</td>
<td>RE</td>
</tr>
<tr>
<td>W1</td>
<td>21</td>
<td>W2</td>
<td>0</td>
<td>RE</td>
</tr>
</tbody>
</table>
**SPECTRUM ANALYSIS WITH FFT**

**INPUT FUNCTION:** \( (x, y) \exp(-y^2) \quad \text{FACTOR:} \quad c_1=1.0000 \quad c_2=1.0000 \)

**NUMBER OF SPECTRUM:** \( N_1=16 \quad N_2=16 \)

**INPUT SAMPLE STEP:** \( t=1.0000 \)

**SPECTRUM \( F(W_1,W_2) \):**

<table>
<thead>
<tr>
<th>( W_1 )</th>
<th>( W_2 )</th>
<th>( R = \text{REAL}(F(W_1, W_2)) )</th>
<th>( I = \text{IMAG}(F(W_1, W_2)) )</th>
<th>( M = \text{MODEL}(F(W_1, W_2)) )</th>
<th>( A = \text{ANGLE}(F(W_1, W_2)) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.703374</td>
<td>0.000000</td>
<td>0.703374</td>
<td>0.000000</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0.074790</td>
<td>-0.057873</td>
<td>0.174693</td>
<td>295.348630</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0.044721</td>
<td>-0.079440</td>
<td>0.091163</td>
<td>299.377260</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0.030854</td>
<td>-0.043685</td>
<td>0.063673</td>
<td>308.025940</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0.036803</td>
<td>-0.033301</td>
<td>0.049633</td>
<td>317.600850</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0.035893</td>
<td>-0.022282</td>
<td>0.038026</td>
<td>328.685490</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0.035425</td>
<td>-0.013822</td>
<td>0.035826</td>
<td>334.319340</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0.035205</td>
<td>-0.006640</td>
<td>0.035139</td>
<td>349.000000</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0.035129</td>
<td>0.000000</td>
<td>0.035139</td>
<td>349.000000</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0.035205</td>
<td>-0.006640</td>
<td>0.035826</td>
<td>349.000000</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0.035425</td>
<td>-0.013822</td>
<td>0.036028</td>
<td>351.154252</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0.035205</td>
<td>-0.022282</td>
<td>0.035826</td>
<td>351.154252</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0.035205</td>
<td>-0.006640</td>
<td>0.035826</td>
<td>351.154252</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0.035129</td>
<td>0.000000</td>
<td>0.035826</td>
<td>351.154252</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0.035205</td>
<td>-0.006640</td>
<td>0.035826</td>
<td>351.154252</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0.035205</td>
<td>-0.006640</td>
<td>0.035826</td>
<td>351.154252</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0.035205</td>
<td>-0.006640</td>
<td>0.035826</td>
<td>351.154252</td>
</tr>
</tbody>
</table>

**TABLE 4** Spectrum Analysis with FFT (\( N_1=16, N_2=16 \))
**TABLE 5 Spectrum Analysis with RFFT (N1=8, N2=10)**
**SPECTRUM ANALYSIS WITH RFFT**

**INPUT FUNCTION:**
\[ f(x, y) = \exp(-x*c_1 + y*c_2) \]

**FACTOR:**
- \( c_1 = 1.0000 \)
- \( c_2 = 1.0000 \)

**NUMBER OF INPUT:**
- \( N_x = 4 \)
- \( N_y = 16 \)

**NUMBER OF SPECTRUM:**
- \( M_x = 16 \)
- \( M_y = 16 \)

**RECURSIVE TIMES:**
- \( H = 4 \)

**INPUT SAMPLE STEPS:**
- \( T = 0.1000 \)

---

**TABLE 6 Spectrum Analysis with RFFT (N1=16, N2=16)**

<table>
<thead>
<tr>
<th>( M_1 )</th>
<th>( M_2 )</th>
<th>( W_1 )</th>
<th>( W_2 )</th>
<th>( RE )</th>
<th>( IM )</th>
<th>( MOD )</th>
<th>( ANL )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.703374</td>
<td>0.000000</td>
<td>0.703374</td>
<td>0.000000</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.074790</td>
<td>-0.157874</td>
<td>0.174693</td>
<td>0.348630</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.044721</td>
<td>-0.079440</td>
<td>0.091163</td>
<td>0.377200</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.038854</td>
<td>-0.049685</td>
<td>0.063073</td>
<td>0.025760</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.036803</td>
<td>-0.033301</td>
<td>0.049633</td>
<td>0.161320</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.035863</td>
<td>-0.022822</td>
<td>0.038026</td>
<td>0.385060</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.035425</td>
<td>-0.013822</td>
<td>0.035826</td>
<td>0.391210</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.035205</td>
<td>-0.006640</td>
<td>0.035139</td>
<td>0.00109</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.035139</td>
<td>0.000000</td>
<td>0.035139</td>
<td>0.00109</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.035205</td>
<td>0.006640</td>
<td>0.035826</td>
<td>0.385060</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.035425</td>
<td>0.013822</td>
<td>0.038026</td>
<td>0.391210</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.035205</td>
<td>0.022822</td>
<td>0.042239</td>
<td>0.314674</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.035863</td>
<td>0.033301</td>
<td>0.049633</td>
<td>0.383631</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.036803</td>
<td>0.033301</td>
<td>0.049633</td>
<td>0.383631</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.038854</td>
<td>0.049685</td>
<td>0.063073</td>
<td>0.025760</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.044721</td>
<td>0.079440</td>
<td>0.091163</td>
<td>0.377200</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.074790</td>
<td>-0.157874</td>
<td>0.174693</td>
<td>0.348630</td>
</tr>
</tbody>
</table>

**TABLE 6 Spectrum Analysis with RFFT (N1=16, N2=16)**
### TABLE 7 Spectrum Analysis with RFFT (N1=20, N2=25)

<table>
<thead>
<tr>
<th>W1</th>
<th>W2</th>
<th>F(W1,W2)</th>
<th>RE=REAL(F(W1,W2))</th>
<th>IM=IMAG(F(W1,W2))</th>
<th>MOD=MODEL(F(W1,W2))</th>
<th>ANL=ANGLE(F(W1,W2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0.074431</td>
<td>0.000000</td>
<td>0.074431</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.123567</td>
<td>0.212920</td>
<td>0.143151</td>
<td>0.212920</td>
<td>0.212920</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0.073167</td>
<td>0.129972</td>
<td>0.143151</td>
<td>0.129972</td>
<td>0.129972</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.043501</td>
<td>0.079177</td>
<td>0.091721</td>
<td>0.079177</td>
<td>0.079177</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0.043658</td>
<td>0.061494</td>
<td>0.066215</td>
<td>0.061494</td>
<td>0.066215</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.056055</td>
<td>0.022614</td>
<td>0.066215</td>
<td>0.022614</td>
<td>0.066215</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0.04438</td>
<td>0.016434</td>
<td>0.052098</td>
<td>0.016434</td>
<td>0.052098</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0.037885</td>
<td>0.000000</td>
<td>0.043795</td>
<td>0.000000</td>
<td>0.043795</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0.04438</td>
<td>0.016435</td>
<td>0.052098</td>
<td>0.016435</td>
<td>0.052098</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0.056055</td>
<td>0.022615</td>
<td>0.066215</td>
<td>0.022615</td>
<td>0.066215</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0.057954</td>
<td>0.016434</td>
<td>0.052098</td>
<td>0.016434</td>
<td>0.052098</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0.04438</td>
<td>0.000000</td>
<td>0.043795</td>
<td>0.000000</td>
<td>0.043795</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0.04438</td>
<td>0.016435</td>
<td>0.052098</td>
<td>0.016435</td>
<td>0.052098</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0.056055</td>
<td>0.022615</td>
<td>0.066215</td>
<td>0.022615</td>
<td>0.066215</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0.043657</td>
<td>0.016434</td>
<td>0.052098</td>
<td>0.016434</td>
<td>0.052098</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0.04438</td>
<td>0.000000</td>
<td>0.043795</td>
<td>0.000000</td>
<td>0.043795</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0.043657</td>
<td>0.016435</td>
<td>0.052098</td>
<td>0.016435</td>
<td>0.052098</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0.04438</td>
<td>0.000000</td>
<td>0.043795</td>
<td>0.000000</td>
<td>0.043795</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0.057954</td>
<td>0.016435</td>
<td>0.052098</td>
<td>0.016435</td>
<td>0.052098</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0.04438</td>
<td>0.000000</td>
<td>0.043795</td>
<td>0.000000</td>
<td>0.043795</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0.057954</td>
<td>0.016435</td>
<td>0.052098</td>
<td>0.016435</td>
<td>0.052098</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0.04438</td>
<td>0.000000</td>
<td>0.043795</td>
<td>0.000000</td>
<td>0.043795</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0.057954</td>
<td>0.016435</td>
<td>0.052098</td>
<td>0.016435</td>
<td>0.052098</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>0.04438</td>
<td>0.000000</td>
<td>0.043795</td>
<td>0.000000</td>
<td>0.043795</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0.057954</td>
<td>0.016435</td>
<td>0.052098</td>
<td>0.016435</td>
<td>0.052098</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>0.04438</td>
<td>0.000000</td>
<td>0.043795</td>
<td>0.000000</td>
<td>0.043795</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0.057954</td>
<td>0.016435</td>
<td>0.052098</td>
<td>0.016435</td>
<td>0.052098</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>0.04438</td>
<td>0.000000</td>
<td>0.043795</td>
<td>0.000000</td>
<td>0.043795</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0.139361</td>
<td>0.212920</td>
<td>0.254039</td>
<td>0.212920</td>
<td>0.254039</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>0.102492</td>
<td>0.233756</td>
<td>0.255232</td>
<td>0.233756</td>
<td>0.255232</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0.040583</td>
<td>0.061856</td>
<td>0.073981</td>
<td>0.061856</td>
<td>0.073981</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>0.026108</td>
<td>0.034713</td>
<td>0.043435</td>
<td>0.034713</td>
<td>0.043435</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>0.015702</td>
<td>0.021608</td>
<td>0.026711</td>
<td>0.021608</td>
<td>0.026711</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>0.003701</td>
<td>0.017093</td>
<td>0.018010</td>
<td>0.017093</td>
<td>0.018010</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0.004701</td>
<td>0.017887</td>
<td>0.017887</td>
<td>0.017887</td>
<td>0.017887</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>0.000442</td>
<td>0.017887</td>
<td>0.017887</td>
<td>0.017887</td>
<td>0.017887</td>
</tr>
</tbody>
</table>

**SPECTRUM ANALYSIS WITH RFFT**

**INPUT FUNCTION:** f(x,y) = exp(-((x^2+y^2)/2))

**FACOR:** e = 1, 0000

**NUMBER OF INPUT; N1=4, N2=25**

**NUMBER OF SPECTRUM; N1=16, N2=16**

**RECURSIVE TIMES: N=5**

**INPUT SAMPLE STEP: T=0.1000**
**SPECTRUM ANALYSIS WITH RFFT**

**INPUT FUNCTION:**
\[ f(x,y) = e^{-(x \cdot c_1 + y \cdot c_2)} \]

**FACTOR:**
- \( c_1 = 1.0000 \)
- \( c_2 = 1.0000 \)

**NUMBER OF INPUT:**
- \( N_x = 8 \)
- \( N_y = 60 \)

**NUMBER OF SPECTRUM:**
- \( H_x = 32 \)
- \( H_y = 32 \)

**RECURSIVE TIMES:**
- \( H = 7 \)

**INPUT SAMPLE STEP:**
- \( T = 0.0500 \)

---

<table>
<thead>
<tr>
<th>W1</th>
<th>W2</th>
<th>RE</th>
<th>IM</th>
<th>MOD</th>
<th>ANL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.93</td>
<td>0</td>
<td>0.000000</td>
<td>6.000000</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0.065763</td>
<td>-0.237266</td>
<td>0.247310</td>
<td>286.384770</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0.042537</td>
<td>-0.127783</td>
<td>0.174677</td>
<td>289.411870</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0.036575</td>
<td>-0.077777</td>
<td>0.085682</td>
<td>294.078200</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0.026763</td>
<td>-0.054986</td>
<td>0.061153</td>
<td>299.064760</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0.024325</td>
<td>-0.047086</td>
<td>0.052995</td>
<td>297.522570</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0.027924</td>
<td>-0.035987</td>
<td>0.047631</td>
<td>302.925800</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0.027667</td>
<td>-0.029057</td>
<td>0.039404</td>
<td>314.599300</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0.024016</td>
<td>-0.022844</td>
<td>0.033146</td>
<td>316.432660</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0.024159</td>
<td>-0.021508</td>
<td>0.032346</td>
<td>318.321960</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0.026769</td>
<td>-0.017246</td>
<td>0.031844</td>
<td>327.207660</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0.025928</td>
<td>-0.011478</td>
<td>0.026355</td>
<td>336.121700</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0.023543</td>
<td>-0.009467</td>
<td>0.025375</td>
<td>338.094660</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0.024570</td>
<td>-0.006904</td>
<td>0.026134</td>
<td>340.683630</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0.026606</td>
<td>-0.005137</td>
<td>0.024999</td>
<td>349.032900</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0.025115</td>
<td>-0.000890</td>
<td>0.025130</td>
<td>357.994140</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0.023445</td>
<td>-0.000002</td>
<td>0.023445</td>
<td>359.301550</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>0.023115</td>
<td>-0.000876</td>
<td>0.025130</td>
<td>1.999240</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0.026507</td>
<td>-0.005133</td>
<td>0.026999</td>
<td>10.959450</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>0.024571</td>
<td>-0.009890</td>
<td>0.02134</td>
<td>19.911145</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0.023544</td>
<td>-0.009464</td>
<td>0.025375</td>
<td>21.899247</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0.023672</td>
<td>-0.017242</td>
<td>0.029355</td>
<td>32.720237</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>0.024325</td>
<td>-0.017242</td>
<td>0.031844</td>
<td>32.720237</td>
</tr>
</tbody>
</table>

**TABLE 8** Spectrum Analysis with RFFT (N1=56, N2=60)
CHAPTER VI

BIBLIOGRAPHY


Kumaresan, R. & Tufts, D.W., "Estimating the parameters of


Preuss, R.D., "Very fast computation of the radix-2 DFT


Stuller,J.A., "Generalized running discrete transform",\)


