AD-A061 030 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO F/G 9/3

DIGITAL SIGNAL INTERPOLATION USING MATRIX TECHNIQUES AND THE WH--ETC(U)

MAY 78 J C WHEELER

UNCLASSIFIED AFIT-CI-78-120D NL

1 OF 3

AD A061030

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CI-78-120D	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Digital Signal Interpolation Using Matrix Techniques and the Whittaker Cardinal Function		5. TYPE OF REPORT & PERIOD COVERED Dissertation
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Joseph C. Wheeler		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS AFIT Student at the University of New Mexico, Albuquerque, New Mexico		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS AFIT/CI WPAFB OH 45433		12. REPORT DATE 1978
		13. NUMBER OF PAGES 190 Pages
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		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

SEP 22 1978

JOSEPH P. HIPPS, Major, USAF
Director of Information, AFIT

APPROVED FOR PUBLIC RELEASE AFR 190-17.

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

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

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

78-120D

(6) DIGITAL SIGNAL INTERPOLATION
USING MATRIX TECHNIQUES
AND THE WHITTAKER CARDINAL FUNCTION

By

(10) Joseph C. Wheeler (11) May 78

B.S., Agricultural and Mechanical
College of Texas, 1961
M.S., Air Force Institute of Technology, 1965

(14) AFIT-CI-78-120D

(9) Doctoral thesis

(12) 206 p.

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering
in the Graduate School of
The University of New Mexico
Albuquerque, New Mexico
May, 1978

ACKNOWLEDGEMENTS

I was priviledged and honored to have Dr. Ahmed Erteza as the Chairman of my Doctoral Committee. His overall impact is reflected in my completing all degree requirements smoothly and within the strict time constraints allotted to me by my sponsor. I am forever indebted to Dr. Erteza for his leadership and personal interest in my program.

Dr. Cleve Moler's impact on this work will be obvious on reading. For most of my professional career, I assumed that because engineers first designed and programmed computers, that they had a corner on computing knowledge. Dr. Moler, a mathematician, shattered my image. The viewing of a computer through the eyes of the numerical analyst was truly a revelation to me.

I am grateful to Dr. Tom Cordaro for being a friend and counselor during this period of study. Whenever I had doubts or questions, Dr. Cordaro invariably provided the encouragement to keep me going or the answers to help me over a technical hurdle.

The initial ideas for my dissertation came while a student of Dr. John Brayer. I am indebted to Dr. Brayer for structuring his course freely enough that I was able to develop some initial ideas about matrices and digital filters into a term paper and ultimately into this dissertation topic.

I am grateful to the other professors, notably Drs. Koschmann and Karni of the Electrical Engineering Department and Dr. Demar of the Math Department, whose influence and guidance helped me through this ordeal. I am also indebted to Dr. Kun-Shan Lin who spent many hours constructively criticizing and discussing my work. Kun-Shan was truly a help in developing my ideas. I must also thank Chris Tolendino for her patient and professional typing of this final manuscript.

I ultimately owe this next formal chapter in my education to my mother and father. I would never have achieved the prerequisite educational plateaus without their start. And to my wife, Joyce, and my son and daughter, Jody and Jaime, I am both eternally grateful and apologetic. Grateful for their love and encouragement--apologetic for the sacrifices they made so that I could devote full time to study. We all share equally in any credit for this endeavor for surely it was neither possible nor worth it without them.

This work was sponsored by the United States Air Force through the Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio.

DIGITAL SIGNAL INTERPOLATION

USING MATRIX TECHNIQUES

AND THE WHITTAKER CARDINAL FUNCTION

By

Joseph C. Wheeler

ABSTRACT OF DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy in Electrical Engineering
in the Graduate School of
The University of New Mexico
Albuquerque, New Mexico

May, 1977

DIGITAL SIGNAL INTERPOLATION USING MATRIX TECHNIQUES AND THE WHITTAKER CARDINAL FUNCTION

Joseph C. Wheeler, Ph.D.
Department of Electrical Engineering
and Computer Science
The University of New Mexico, 1977

ABSTRACT

The infinite Whittaker summation and Shannon's sampling theorem both use the weighted sum of sinc functions (the "Cardinal" function) in the interpolation algorithm. When the number of original samples is approximately equal to twice the product of duration (T) and bandwidth (W), and when it is desired to increase the number of samples by powers of 2, the interpolation process can be written as a matrix equation. It is shown that when the original sample set is periodic, the matrix elements converge to simple cosecant and cotangent functions.

The "Whittaker" matrix developed for either the 2TW transient or the periodic sample sets can be manipulated into a real symmetric matrix format. It is shown that a unitary equivalence transformation on the periodic matrix implemented via the Fast Fourier Transform (FFT) and an orthogonal similarity transformation on the symmetric matrix are really equivalent algorithms. It is also shown that by suitably modifying the transient Whittaker matrix, an orthogonal similarity transformation is possible which can significantly reduce the number of computations necessary to interpolate a data set. When "a-priori" knowledge about a data set is not available, but it is desired to fit a specific curve to the data, it is shown that the eigenvectors of the modified matrix are a unique orthogonal basis for the space which includes the data set; linear combinations of the basis vectors can then be made using a technique called the Eigen-Filter, Cross-Correlation Algorithm to indicate how well the combination works. Numerous examples are given which show that this algorithm can provide a better interpolation than Fourier techniques.

Matrix norms and condition numbers are used to bound truncation errors, computer round-off errors, and errors due to the curve fitting algorithm. It is also shown that when "noisy" data is interpolated, the noise-to-signal ratio of the interpolants can be magnified by the interpolating matrix condition number.

An extensive computer program which implements the algorithms is described. Numerous signals are processed and the results presented in plots and tabular form. The work is ended with an entire chapter suggesting areas for follow-on work.

LIST OF FIGURES

LIST OF SYMBOLS

$\alpha\,\sqcap_{\beta}\,(t)$	Rectangular shaped waveform, amplitude α, pulse width β, and a function of t. Subscript "n" instead of "(t)" indicates discrete function.
$\alpha\,\sqcap_{\beta_1}{}^{\beta_2}\sqcap_{\beta_3}\,(t)$	Rectangular shaped waveform, amplitude α, first pulse width β_1, distance to second pulse β_2, second pulse width β_3, and a function of t. Subscript "n" instead of "(t)" indicates discrete function.
$\alpha\,\sqcap_{\beta}\,(t)_p = \sum_{k=-\infty}^{\infty} \alpha\,\sqcap_{\beta}\,(t-kT_s)$	Periodic extension of $\alpha\,\sqcap_{\beta}\,(t)$. Repetition period = T_s.
$C(t) = \sum_{k=-\infty}^{\infty} f(kT_s) \dfrac{\mathrm{SIN}\frac{\pi}{T_s}(t-kT_s)}{\frac{\pi}{T_s}(t-kT_s)}$	Whittaker cardinal function.
$\delta_p(t) = \sum_{k=-\infty}^{\infty} \delta(t-kT_s)$	Periodic "train" of delta functions; Repetition period = T_s.
f	Vector of interpolants.
f_i	"i^{th}" scalar component of vector f.
$f^N = P^N x^N$	Periodic or partial periodic interpolating equation. P is NxN, f is Nxl, and x is Nxl. Superscript indicates dimension, NOT power.
$f^{(i)} = P^i f^{(0)}$	Recursive periodic interpolating equation. "(i)" indicates "i^{th}" iteration. Superscript on P indicates power: $f^{(1)}=Pf^{(0)}$, $f^{(2)}=P^2f^{(0)} \ldots f^{(N)}=P^Nf^{(0)}$.

g	Vector f written in reverse order.
g_i	"i^{th}" scalar component of vector g.
$g^N = S^N x^N$	Symmetric interpolating equation. S is a symmetric matrix, $S=S^T$. Superscripts have same interpretation as for $f^N=P^N x^N$.
$g^{(i)} = S^i x^{(0)}$	Recursive symmetric interpolation equation.
$g = \sum_{j=1}^{N} \lambda_j T v^j v^{j*} x$	Rank 1 matrix decomposition of periodic interpolating equation. λ_j is the "j^{th}" eigenvalue of P, T is the permutation matrix, Tv^j is v^j in reverse order and v^j is the "j^{th}" eigenvector of P, v^{j*} is the complex conjugate transpose of the "j^{th}" eigenvector, x is the data vector.
$g = \sum_{j=1}^{N} \gamma_j u^j u^{jT} x$	Rank 1 matrix decomposition of symmetric interpolating equation. Same interpretation as above except u^{jT} is simple transpose of "j^{th}" eigenvector of S and γ_j is "j^{th}" eigenvalue of S.
$\langle x, y\rangle = x^T y = \sum_{i=1}^{N} x_i y_i$	Inner product notation.
$k(s) = \left\|\frac{\gamma_{max}}{\gamma_{min}}\right\|$	Condition number of symmetric matrix S. Equal to ratio of maximum to minimum eigenvalues of S.
$\|\|\cdot\|\|_p$	"p"-norm of vector or matrix ".". Defined in Section 1, Chapter 7.
$P = [p_{ij}]$	Periodic Whittaker matrix with components p_{ij}.

$P = [p_\ell]$	Periodic Whittaker matrix with cyclical components p_ℓ.
$Q = [q^1 \mid q^2 \mid \cdots \mid q^N]$	Partitioned matrix consisting of N columns of vectors q^j.
q_i^j	"i^{th}" component of "j^{th}" vector.
$S = [s_{ij}]$	Symmetric periodic Whittaker matrix with components s_{ij}.
$S = [s_\ell]$	Symmetric Whittaker matrix with cyclical components s_ℓ.
$S = TV\Lambda V^*$	Unitary equivalence transformation of periodic Whittaker matrix. T is an NxN permutation matrix, V is NxN matrix of complex eigenvectors of P, $VV^* = V^*V = I$, Λ is the NxN diagonal matrix of complex eigenvalues of P.
$S = U\Gamma U^T$	Real decomposition of symmetric matrix. U is NxN matrix of real eigenvectors of S, $UU^T = U^TU = I$, Γ is the NxN diagonal matrix of real eigenvalues of S.
$s_\ell^{(i)}$	"ℓ^{th}" component after "(i)" iterations.
x	Data vector to be interpolated.
x_i	"i^{th}" component of vector x.

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

Interpolation is a well established branch of mathematics. For the engineer, however, the tools to implement interpolation are sometimes lacking; we try to get along with simple, linear approximations because of the low cost and ease of implementation. Because today's mass data acquisition systems dictate minimum sampling rates, these "straight line" approximations are often inadequate. It is our purpose, then, to offer an alternative. We present algorithms which can be implemented on big or small computers, as well as in hardware; but, as opposed to linear "approximations," our algorithms are "exact" when the original phenomenon and its sample set satisfy certain requirements.

Generally, interpolating algorithms are concerned with fitting an "N^{th}" degree polynomial through a set of equally or unequally spaced sample points from some continuous process. It is well known theoretically, that N + 1 sample points can be connected by the curves which plot all polynomials of degree N or greater. E. T. Whittaker set about finding out if any one of all the functions that can be made to fit a data set had any distinguishing properties--"a function of royal blood whose distinguished properties set it apart from its bourgeois brethern." The well known cardinal function (sum of weighted sinc functions) resulted from his work and has the property that it contains no frequencies higher than twice the sampling frequency. In more recent years, the independent but equivalent Shannon sampling theorem was developed and is the basis for much of today's communication work. This theory also derives the sum of weighted sinc functions, but as the consequence of an ideal filter acting on a sequence of data samples from a band-limited signal.

Our approach to interpolation is to formulate the sum of weighted sinc functions into a matrix-vector product. We show that the matrix with elements formed from the Whittaker cardinal function can be written as a symmetric matrix with many useful properties. These properties are what we exploit. The rationale for this approach is that matrices are physical entities; any manipulation of these arrays is directly translatable to computer language or hardware.

1.1 Organization of Work

The remainder of our work is divided into several chapters wherein various aspects of the interpolation problem are discussed. Each chapter is complete in that any derivations begin and end there; the Appendix is reserved for the various computer programs rather than more detailed derivations which this author can better handle in the main text. Symbology is sometimes complicated, so a list of symbols is provided at the beginning of this dissertation to aid the reader.

We begin in Chapter II by reviewing the mathematical process of interpolation in terms of the digital filter. The engineering "band-limited" function is introduced and subsequently, its sampled data version is generated by first sampling with a sequence of delta functions, and then with a more realistic sequence of rectangular waveforms. It is this sample data set that we wish to interpolate. We show that if the samples are passed through an appropriate low-pass digital filter, the resulting continuous waveform is a reconstructed version of the original function. Practically, a more dense data set rather than a continuous function is the object of the interpolation scheme. Two finite interpolation schemes are discussed: Interpolation by the Discrete Fourier Transform, and a time domain convolution of the truncated Whittaker cardinal function and the sample data set.

In Chapter III, we present an historical review of interpolation involving the weighted sum of sinc functions. We present the work of E. T. Whittaker [36] wherein he developed the famous cardinal function, and we follow through with the work of his son, J. M. Whittaker [37], who showed what types of sample sets generate the cardinal function. The communications sampling theorem developed independently from Whittaker is discussed again but now from the point of view of how it relates to Whittaker. We present brief reviews of several important papers by Hartley [14], Nyquist [21], and Shannon [30] which emphasize the importance of how many samples are needed to generate the cardinal function.

The original effort in this dissertation begins with Chapter IV wherein we select one of many possible interpolation intervals and formulate the interpolating equations as products of matrices and sample data vectors. The choice of the "mid-point" interpolation interval and square interpolating matrix may seem arbitrary, but we show later that once this problem is mastered, the number of samples can be successively doubled by recursively applying the mid-point algorithm. We also introduce the concepts of periodic and transient matrix operations; i.e., we develop a special matrix to implement the mid-point algorithm for band-limited, periodic, time domain signals, and develop a separate truncated matrix for band limited, time domain transients. The two forms of the Whittaker matrix are "massaged" until they are symmetric and the matrix elements can be expressed in a closed form trigonometric expression for the periodic case, or as a sequence of terms from a truncated infinite series for the transient case.

In Chapter V we show that a matrix expression for a periodic convolution process has very special properties [13], [15]. Specifically, the matrix is circulant and can be decomposed into the product of a complex unitary matrix with a matrix of complex eigenvalues, followed by a product with the complex conjugate transpose of the unitary matrix. The eigenvalues are determined from the Fourier Transform of one of the rows of the circulant matrix, and the elements of the unitary matrices are themselves samples from the Fourier kernel. We explore this decomposition for the periodic Whittaker matrix in cyclic form; we derive closed form expressions for the matrix eigenvalues and prove the surprising result that the even ordered periodic Whittaker matrix is singular. We conclude the chapter by describing an interpolation algorithm which uses the Fast Fourier Transform (FFT).

Both the periodic and transient Whittaker matrices in real symmetric form are orthogonally similar to a diagonal matrix of real eigenvalues. In Chapter VI we describe a computer technique based on the Francis QR [9], [20] algorithm to generate the orthogonal matrices of eigenvectors and the diagonal matrix of eigenvalues. We then show that this decomposition can be viewed as a discrete cross-correlation process, and if certain properties are known to be present in the sample set, significant savings can be achieved over straightforward implementation of the matrix products.

The purpose of Chapter VII is to present error bounds for the interpolation algorithms. Four types of errors are discussed: series truncation errors, errors due to noisy data, machine round off errors, and errors caused by the Eigen-Filter, Cross-Correlation algorithm described in Chapter VI. The benefits of casting the interpolation problem as a matrix process become evident in this chapter when we discuss the error bounds in terms of matrix norms and condition numbers.

In Chapter VIII we present the results of our work in the form of plots and graphs of the outputs from the various algorithms. We show that the major algorithms as programmed in the Appendix do work and are practical. We also summarize our goals and findings as a conclusion to our work.

Chapter IX is a special chapter wherein we outline other problems associated with interpolation. First, because of the mid-point interpolation scheme and square interpolating matrix, N - 1 interpolants are actually computed and one extrapolant is produced. If the interpolants themselves are interpolated, N - 2 of the original N points are returned and two extrapolants produced. Can this process be continued? The inverse interpolation problem is also discussed. In particular, given the interpolants, how do we compute the original data vector? This problem is not straightforward when the periodic Whittaker matrix is even ordered; i.e., the matrix is singular. We also outline a fast way for computing the derivative of the original sample set. This can be accomplished for the periodic case via a decomposition of a modified Whittaker matrix implemented with the FFT. Finally, we discuss a recursive algorithm which would allow interpolating intervals other than the mid point to be approached.

The final parts of the dissertation are a compendium of computer programs in the Appendix, and a Bibliography.

CHAPTER II

ENGINEERING APPROACH TO INTERPOLATION

In Section 1 of this chapter, we discuss two popular sampling functions used on continuous waveforms: first, we present the idealized impulse train used for theoretical work and then we develop the realistic rectangular pulse function. We introduce the dual frequency - time relationships of these functions, and in Section 2 we apply the sampling functions to engineering "band-limited" signals to produce the sample data set. Next, we show the periodic nature of the Fourier spectrum of this sample data set, and then we prove that when such a spectrum is filtered with an ideal low-pass filter, the original continuous time domain signal is returned.

When a continuous signal is not needed from the sample set, then a more dense data set may be the object of an interpolation scheme. Sections 3 and 4 are two approaches to this problem. In Section 3 we present the Discrete Fourier Transform (DFT) interpolator--basically, a frequency domain technique, and in Section 4 we present a time domain convolution approach to interpolation.

The material in this chapter is covered in numerous textbooks and papers. It is presented here for the sake of completeness in discussing the interpolation problem. Particulary useful references for the first two sections are books by Stanley [31, Chapter 3] and Brigham [3, Chapter 6]. In particular, we follow Stanley's lucid development of the sampling functions in Section 1 and use Brigham's pictorial approach to the sampling theorem and reconstruction in Sections 2 and 3. Oppenheim's book [23, Chapter 3, problem 21] and the papers by Schaefer [27, pp. 692-702], Urkowitz [35, pp. 146-154], Oetken [22, pp. 301-309], Crochiere [6, pp. 444-456], and Rabiner [26, pp. 457-464] provided the motivation for Section 3. Stearn's book [32] provided an overall reference for sampling and reconstruction and was particularly important because it was the text for two of this author's signal processing courses at the University of New Mexico.

2.1 Sampling Functions

The well known complex Fourier series representation for a periodic function is expressed by Stanley [31, p. 38]

$$x_p(t) = \sum_{m=-\infty}^{\infty} c_m e^{j2\frac{\pi m t}{T}} \qquad (2\text{-}1)$$

where

$$c_m = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) e^{-j2\frac{\pi m t}{T}} dt \qquad (2\text{-}2)$$

The period T is the interval over which the function completes one cycle of its periodicity, and the c_m are the complex

Fourier coefficients of the waveform. The distinction between $x_p(t)$ and $x(t)$ is that $x(t)$ is one cycle of a periodic train of cycles, i.e.,

$$x_p(t) = \sum_{k=-\infty}^{\infty} x(t - kT) \qquad (2\text{-}3)$$

with

$$x(t) = \begin{cases} x_p(t) & -\frac{T}{2} \le t \le \frac{T}{2} \\ 0 & |t| > \frac{T}{2} \end{cases} \qquad (2\text{-}4)$$

Parallel to the Fourier series representation of a signal is its Fourier transform

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt \qquad (2\text{-}5)$$

with

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df \qquad (2\text{-}6)$$

$X(f)$ is known as the Fourier spectrum of $x(t)$. Of particular interest here, is that when $x(t)$ is a pulse ($x(t) = 0$, $|t| > \frac{T}{2}$) the Fourier coefficients of the <u>periodic</u> extension of $x(t)$, $x_p(t)$, are simply given by

$$c_m = \frac{X(\frac{m}{T})}{T} \qquad (2\text{-}7)$$

One sampling function widely used in theoretical work is the ideal infinite impulse train consisting of unit delta functions at intervals of T_s extending to $\pm\infty$. For such a waveform, we can write symbolically

$$\delta_p(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT_s) \qquad (2\text{-}8)$$

Such a sampling function, being periodic, has a Fourier series expansion. From equation 2-6 we write for a single delta function

$$X(f) = \int_{-\frac{T}{2}}^{\frac{T}{2}} \delta(t)e^{-j2\pi ft} dt = 1 \qquad (2\text{-}9)$$

then from equation 2-7

$$c_m = \frac{x(f)}{T_s} = \frac{1}{T_s} \qquad (2\text{-}10)$$

and substituting in equation 2-1

$$\delta_p(t) = \frac{1}{T_s} \sum_{m=-\infty}^{\infty} e^{j2\frac{\pi mt}{T_s}} \qquad (2\text{-}11)$$

The Fourier spectrum of equation 2-11 itself is given by

$$\Delta(f) = \int_{-\infty}^{\infty} \frac{1}{T_s} \sum_{m=-\infty}^{\infty} e^{-j2\pi(f - \frac{m}{T_s})t} dt \qquad (2\text{-}12)$$

which is well known [3, p. 22]

$$\Delta(f) = \frac{1}{T_s} \sum_{m=-\infty}^{\infty} \delta(f - \frac{m}{T_s}) \tag{2-13}$$

In other words, the spectrum of a periodic impulse sampling train is itself a periodic impulse train.

In the real world, the ideal impulse train cannot be generated. However, the rectangular pulse train can be realized. Proceeding as we did for the impulse train, we write (using symbolic notation)

$$\sqcap_{\alpha}{}_p(t) = \sum_{k=-\infty}^{\infty} \frac{1}{\alpha} \sqcap_{\alpha}(T - kT_s) \tag{2-14}$$

Then, for a single pulse

$$\text{sinc}(f) = \int_{-\infty}^{\infty} \frac{1}{\alpha} \sqcap_{\alpha}(t)\; e^{-j2\pi ft} dt \tag{2-15}$$

$$= \frac{\sin \pi f \alpha}{\pi f \alpha} \tag{2-16}$$

The Fourier coefficients become

$$c_m = \frac{1}{T_s} \frac{\sin(\frac{\pi m \alpha}{T_s})}{(\frac{\pi m \alpha}{T_s})} \tag{2-17}$$

and the Fourier series is

$$\sqcap_{\alpha}{}_p(t) = \frac{1}{T_s} \sum_{m=-\infty}^{\infty} \frac{\sin(\frac{\pi m \alpha}{T_s})}{(\frac{\pi m \alpha}{T_s})} e^{j2\frac{\pi m t}{T_s}} \tag{2-18}$$

The Fourier spectrum of 2-18 is then found as

$$\mathrm{sinc}\left(\frac{m}{T_s}\right) = \int_{-\infty}^{\infty} \frac{1}{T_s} \sum_{m=-\infty}^{\infty} \frac{\sin\left(\frac{\pi m \alpha}{T_s}\right)}{\left(\frac{\pi m \alpha}{T_s}\right)} e^{-j2\pi\left(f - \frac{m}{T_s}\right)t} dt \tag{2-19}$$

$$\mathrm{sinc}\left(\frac{m}{T_s}\right) = \frac{1}{T_s} \sum_{m=-\infty}^{\infty} \frac{\sin\left(\frac{\pi m \alpha}{T_s}\right)}{\left(\frac{\pi m \alpha}{T_s}\right)} \delta\left(f - \frac{m}{T_s}\right) \tag{2-20}$$

Again, the spectrum of the rectangular pulse train is the ideal periodic impulse train but amplitude modulated by the sinx/x function.

This process of postulating a sampling function, writing its Fourier series expansion, and finding the spectrum, could be carried out for most any sampling waveform. In any case, we should arrive at the form for the Fourier series and spectrum

$$x_p(t) = \frac{1}{T_s} \sum_{m=-\infty}^{\infty} X\left(\frac{m}{T_s}\right) e^{j2\frac{\pi m t}{T_s}} \tag{2-21}$$

where

$$X\left(\frac{m}{T_s}\right) = X(f)\Big|_{f=\frac{m}{T_s}} = \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) e^{-j2\pi f t} dt\Big|_{f=\frac{m}{T_s}} \tag{2-22}$$

and

$$X_p(f) = \frac{1}{T_s} \sum_{m=-\infty}^{\infty} X\left(\frac{m}{T_s}\right) \delta\left(f - \frac{m}{T_s}\right) \tag{2-23}$$

These important properties of sampling functions (equations 21 and 23) are used in the next section.

2.2 Generating the Sample Data Set

One of the fundamental properties of spectral analysis is the duality of time domain multiplication and frequency (spectral) domain convolution; i.e.,

$$h(t) \cdot g(t) \Longleftrightarrow H(f)*G(f) = \int_{-\infty}^{\infty} H(u)G(f-u)du \qquad (2\text{-}24)$$

where "$\Longleftrightarrow$" implies two-way equivalence or duality. This property allows us to determine the effects of sampling a continuous time domain signal g(t) by multiplying by a sampling function of the form of equation 2-21

$$g(t) \cdot x_p(t) = \frac{1}{T_s} \sum_{m=-\infty}^{\infty} g(t) X(\frac{m}{T_s}) e^{j2\frac{\pi m t}{T_s}} \qquad (2\text{-}25)$$

But, from 2-23 and 2-24, this is equivalent to the convolution

$$\int_{-\infty}^{\infty} G(u) X_p(f-u)du = \frac{1}{T_s} \int_{-\infty}^{\infty} G(u) \sum_{m=-\infty}^{\infty} X(\frac{m}{T_s}) \delta(f-\frac{m}{T_s}-u)du \qquad (2\text{-}26)$$

$$= \frac{1}{T_s} \sum_{m=-\infty}^{\infty} X(\frac{m}{T_s}) G(f-\frac{m}{T_s}) \qquad (2\text{-}27)$$

Simply stated, the spectrum of the sampled waveform is the periodic extension of the spectrum of the original continuous waveform, repeated at intervals of $\frac{m}{T_s}$, $|m| = 0, 1, \ldots \infty$. These ideas are summarized in Figure 2-1 where we show an

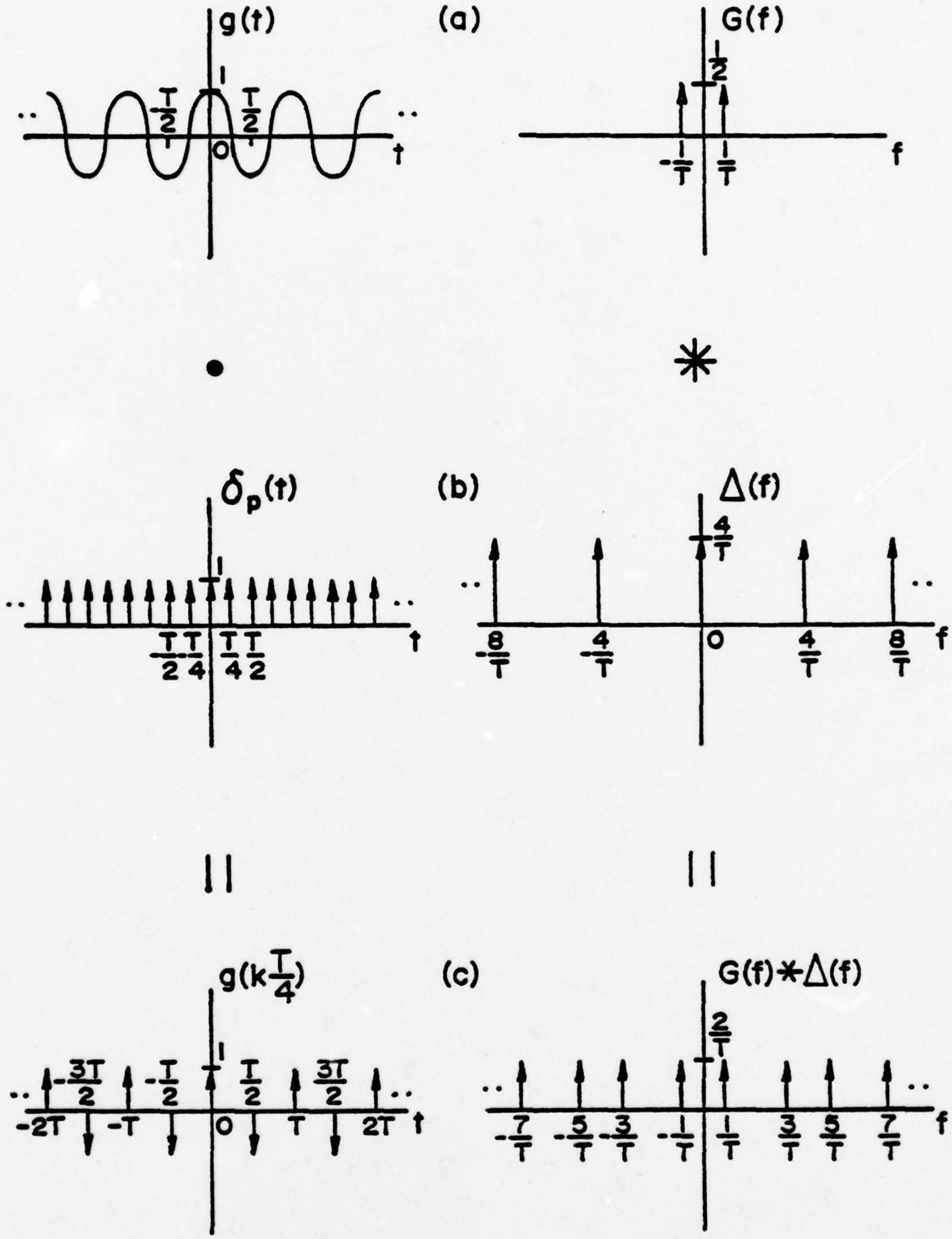


Figure 2-1

Sampling with the Infinite Impulse Train infinite cosine wave sampled by the ideal impulse train. The left side of the figure shows the time domain operations.

The concept of "band-limitedness" is naturally introduced in Figure 2-2(a) where we observe that G(f) is no longer zero for $|f| \geq f_B$. The solid curve for $G_a(f)$ in Figure 2-2c shows that because G(f) is not band-limited, $G_a(f)$ is the sum of the original spectrum G(f) plus the "tails" from all the shifted versions of G(f). This phenomenon, of course, is given the name "aliasing."

Real signals begin and end in finite time and real sampling schemes are of finite duration. These concepts are introduced with the example shown in Figure 2-3. Suppose we theoretically sample this g(t) with an ideal impulse train and then convert to a "pseudo" real world digital process by following the sampling with a rectangular window function. The resulting sample data set is shown in Figure 2-3(e), and can be expressed as

$$g(kT_s)\cdot\omega(t) = \sum_{k=-\infty}^{\infty} [g(t)\cdot\delta(t - kT_s)] \sqcap_{NT_s}\left(t+\frac{T_s}{2}\right) \qquad (2\text{-}28)$$

$$= \sum_{k=0}^{N-1} g(t)\cdot\delta_p(t - kT_s) \qquad (2\text{-}29)$$

which has the equivalent Fourier amplitude spectrum shown in Figure 2-3(e)

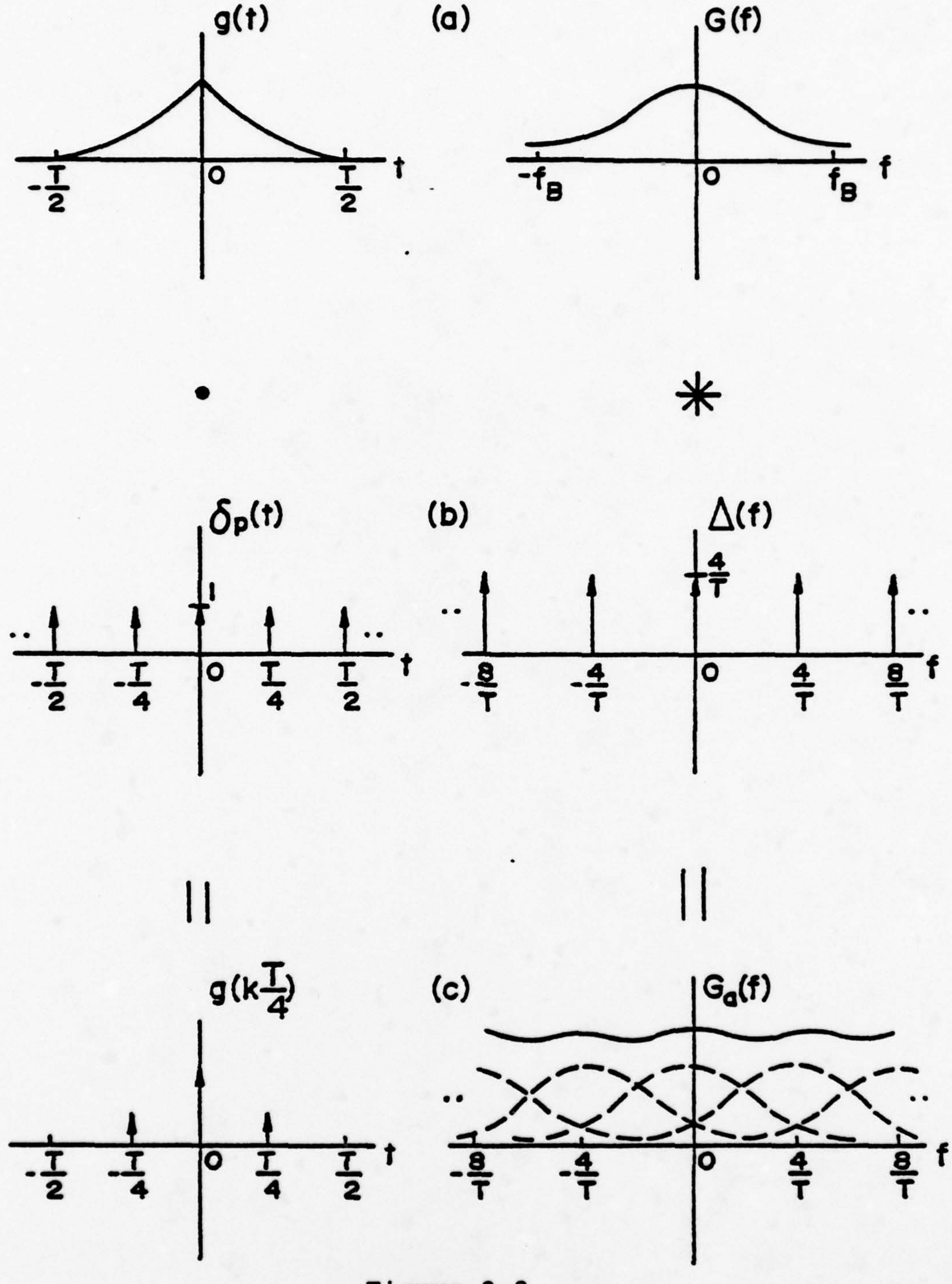


Figure 2-2

Aliasing

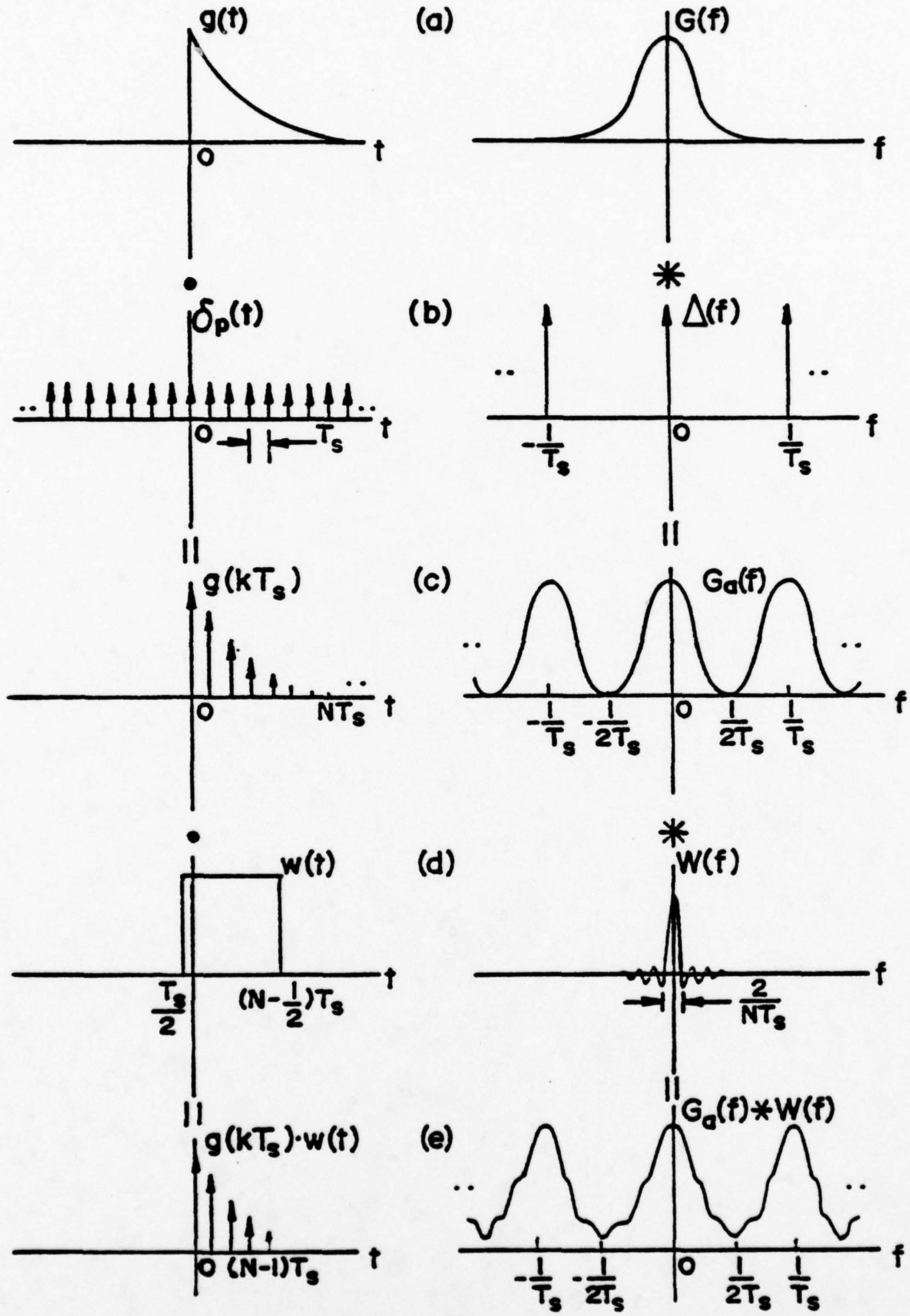


Figure 2-3

Real World Sampling

$$|G_a(f) * W(f)| = \left| \frac{1}{T_s} \sum_{m=-\infty}^{\infty} G_a\left(f - \frac{m}{T_s}\right) * \frac{\sin(\pi f N T_s)}{(\pi f N T_s)} \right| \tag{2-30}$$

Any sampling function could have been used in place of the ideal impulse train. The effects on Figure 2-3 would be that the amplitude of the repeated, modified, spectrum in 2-3(e) would vary as the amplitude of the modulating function in equation 2-27.

The intent of an interpolation scheme is to reconstruct the original waveform from the sample set. Obviously, if we multiply the spectrum in Figure 2-1(c) by the ideal low-pass filter response

$$F(f) = \frac{T}{4} \,\_\!\!\lceil 4/T \rceil\!\!\_\,(f) \tag{2-31}$$

we get back G(f) in Figure 2-1(a). This process is equivalent to passing $g(kT_s)$ through the ideal low-pass filter which returns g(t), the original, infinite duration time domain signal.

The problem is not quite so simple in the case of Figure 2-2(c) where multiplication of the spectrum by the ideal low-pass filter response returns a corrupted version of G(f). The equivalent time domain process of passing the finite length data set through the ideal filter returns a corrupted version of g(t). To overcome this problem, we must increase the sampling rate to conform with the sampling theorem--samples must be taken at a rate at least twice the highest frequency in g(t), or at least a rate such that aliasing is negligible.

In the case of Figure 2-3(e), multiplication of the spectrum by the ideal filter response would have to be followed by a deconvolution process to remove the effects of the windowing operation. Alternatively, we might try and choose windows which have negligible effects on $G_a(f)$. Then the ideal filter essentially returns the original function.

2.3 DFT Interpolation

Implied in all the derivations thus far is the "continuous" nature of time and frequency. But this is not quite satisfactory to explain data manipulation on a computer--we need a completely digital (discrete) version of the dual, frequency-time relationship of discrete data sets. This we provide by extending Figure 2-3(e).

First, sample the spectrum in Figure 2-3(e) (repeated in Figure 2-4(a)) with an infinite impulse train with pulses spaced $1/NT_s$ apart. The equivalent time domain process is convolution with another impulse train, and the overall results are two discrete periodic sequences.

This process can be written as follows:

$$g(kT_s)\cdot\omega(t) = \sum_{k=0}^{N-1} g(t)\,\delta(t - kT_s) \qquad (2\text{-}32)$$

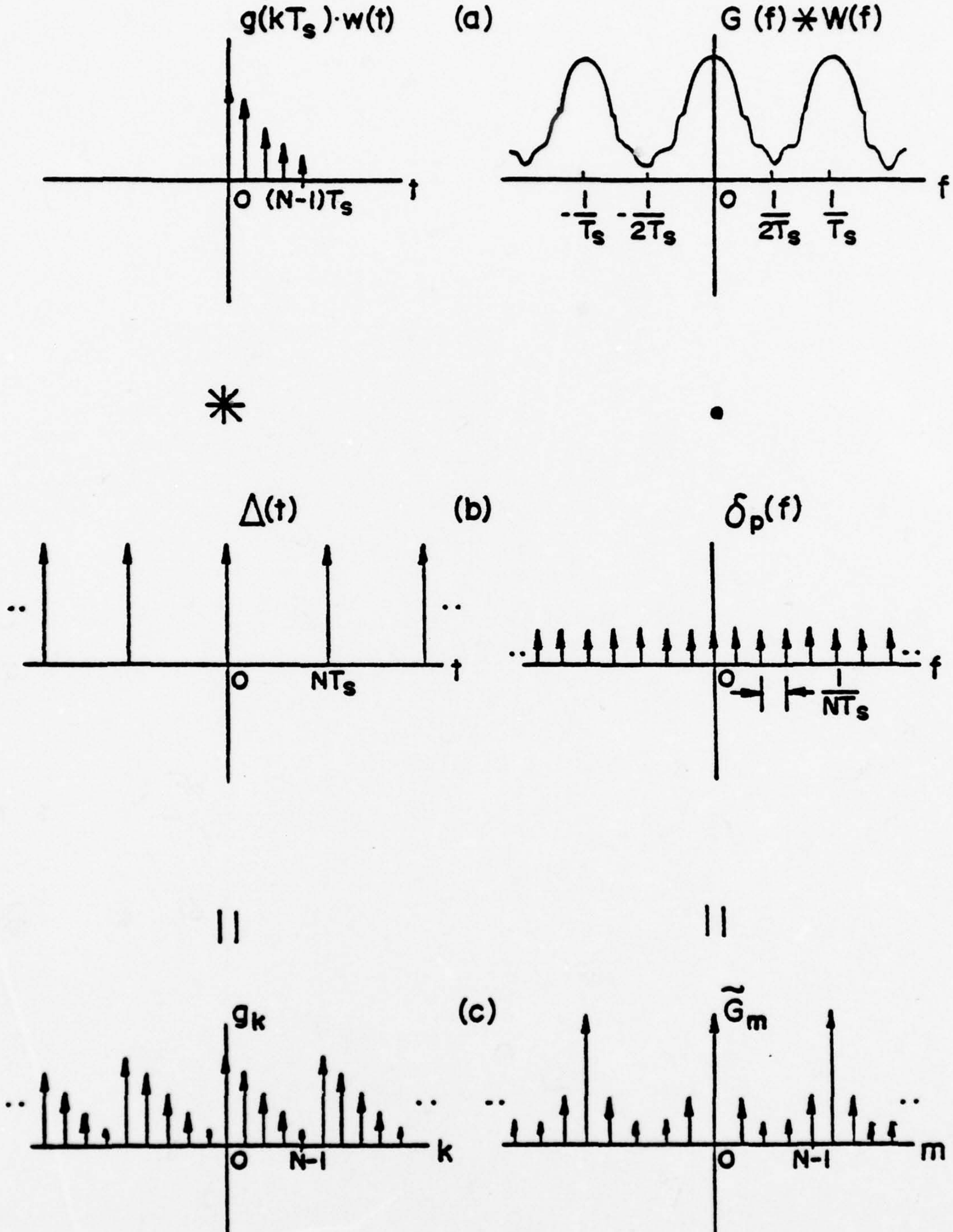


Figure 2-4

Discrete Fourier Transform Pair

$$G_a(f)*W(f) = \int_{-\infty}^{\infty} \sum_{k=0}^{N-1} g(t)\,\delta(t - kT_s)\,e^{-j2\pi ft}\,dt \qquad (2\text{-}33)$$

$$= \sum_{k=0}^{N-1} g(kT_s)\,e^{-j2\pi fkT_s} \qquad (2\text{-}34)$$

Evaluate at $f = \frac{m}{NT_s}$ and we have the periodic Discrete Fourier Transform (DFT) shown in Figure 2-4(c). Now we write for the fundamental period

$$\tilde{G}_m = \sum_{k=0}^{N-1} g(kT_s)\,e^{-j2\frac{\pi km}{N}}, \quad m = 0,\ 1,\ \dots\ N-1 \qquad (2\text{-}35)$$

We can also write the Fourier series for the periodic time domain sample set using equations 2-1 and 2-7

$$[g(kT_s)\cdot\omega(t)]*\Delta(t) = \sum_{m=-\infty}^{\infty} \frac{\tilde{G}_m}{NT_s}\,e^{j2\frac{\pi mk}{N}} \qquad (2\text{-}36)$$

Then, the fundamental period is

$$g_k = \frac{1}{NT_s} \sum_{m=0}^{N-1} \tilde{G}_m\,e^{j2\frac{\pi km}{N}}, \quad k = 0,\ 1,\ \dots,\ N-1 \qquad (2\text{-}37)$$

Equations 2-35 and 2-37 are a Discrete Fourier Transform pair except for the constant $\frac{1}{T_s}$ in equation 2-37. This can be seen by substituting 2-35 into 2-37 (without the $\frac{1}{T_s}$).

$$g_p = \frac{1}{N} \sum_{m=0}^{N-1} \left\{ \sum_{k=0}^{N-1} g_k e^{-j2\frac{\pi km}{N}} \right\} e^{j2\frac{\pi pm}{N}} \tag{2-38}$$

$$= \frac{1}{N} \sum_{m=0}^{N-1} \sum_{k=0}^{N-1} g_k e^{-j2\frac{\pi (k-p)m}{N}} \tag{2-39}$$

$$= \frac{1}{N} \sum_{m=0}^{N-1} g_p \equiv g_p \tag{2-40}$$

The interpolation problem requires that we increase N in the above equations. To increase N by a factor of 2, we proceed as follows: Define a new sequence

$$f_\ell = \left\{ \begin{array}{ll} g_k & \ell = 2k \\ 0 & \ell = 2k+1 \end{array} \right\} k = 0, 1, \ldots N-1 \tag{2-41}$$

Then, equation 2-35 becomes

$$\tilde{F}_m = \sum_{\ell=0}^{2N-1} f_\ell e^{-j2\frac{\pi \ell m}{2N}}, \; m = 0, 1, \ldots, 2N-1 \tag{2-42}$$

$$\tilde{F}_m = \sum_{k=0}^{N-1} f_{2k} e^{-j4\frac{\pi km}{2N}}, \; m = 0, 1, \ldots, 2N-1 \tag{2-43}$$

$$\tilde{F}_m = \sum_{k=0}^{N-1} g_k e^{-j2\frac{\pi km}{N}}, \; m = 0, 1, \ldots, 2N-1 \tag{2-44}$$

which has twice as many spectral samples as does $\tilde{G}_m$. Now multiply F_m by the ideal symbolic filter function which zeroes coefficients from $\frac{N}{2}$ to $\frac{3N}{2}$ and has gain 2, and take the inverse DFT of the product

$$f_\ell = \frac{1}{2N}\sum_{m=0}^{2N-1}\tilde{F}_m \; 2\sqcap_{\frac{N}{2}}\sqcup_{N}\sqcap_{\frac{N}{2}}\,m \; e^{j2\frac{\pi \ell m}{2N}} \tag{2-45}$$

$$= \frac{2}{2N}\sum_{m=0}^{\frac{N}{2}-1}\tilde{F}_m e^{j2\frac{\pi \ell m}{2N}} + \frac{2}{2N}\sum_{m=\frac{3}{2}N+1}^{2N-1}\tilde{F}_m e^{j2\frac{\pi \ell m}{2N}} \tag{2-46}$$

Let r = 2N - m in the second summation; then,

$$f_\ell = \frac{1}{N}\sum_{m=0}^{\frac{N}{2}-1}\tilde{F}_m e^{j2\frac{\pi \ell m}{2N}} + \frac{1}{N}\sum_{r=\frac{N}{2}-1}^{1}\tilde{F}_{2N-r} e^{-j2\frac{\pi \ell r}{2N}} \tag{2-47}$$

From equation 2-44

$$\tilde{F}_{2N-r} = \tilde{F}_r{}^* = \tilde{F}_{-r} \tag{2-48}$$

Then,

$$f_\ell = \frac{1}{N}\sum_{m=0}^{\frac{N}{2}-1}\tilde{F}_m e^{j\frac{\pi \ell m}{N}} + \frac{1}{N}\sum_{m=1}^{\frac{N}{2}-1}\tilde{F}_m{}^* e^{-j\frac{\pi \ell m}{N}} \tag{2-49}$$

Finally,

$$f_\ell = \frac{1}{N}\sum_{m=-(\frac{N}{2}-1)}^{\frac{N}{2}-1}\tilde{F}_m e^{j\frac{\pi \ell m}{N}}, \quad \ell = 0, 1, \ldots, 2N-1 \tag{2-50}$$

Thus, midpoint interpolation is achieved by computing a modified inverse DFT as prescribed by equation 2-50. The zeroing of spectral coefficients used to arrive at equation 2-50 is equivalent to multiplying the periodic spectrum F_m by the spectrum of the sampled version of an ideal lowpass filter.

As an example, consider the sample set generated by a periodic sine wave,

$$g_k = \sin 2\frac{\pi k}{4}\ , \ k = 0,\ 1,\ 2,\ 3 \tag{2-51}$$

Then,

$$\tilde{F}_m = \sum_{k=0}^{3} \sin 2\frac{\pi k}{4} = 0 + e^{-j2\frac{\pi m}{4}} + 0 - e^{-j6\frac{\pi m}{4}} \tag{2-52}$$

$$= 2e^{-j(2m-1)\frac{\pi}{2}} \sin\frac{m\pi}{2},\ m = 0,\ 1,\ \ldots,\ 7 \tag{2-53}$$

Now, using equation 2-50

$$f_\ell = \frac{1}{4} \sum_{m=-1}^{1} 2e^{-j(2m-1)\frac{\pi}{2}} \sin\frac{m\pi}{2}\, e^{j\frac{\pi \ell m}{4}},\ \ell = 0,\ 1,\ \ldots,\ 7 \tag{2-54}$$

$$= \frac{1}{2}\left\{-e^{j3\frac{\pi}{2}}\, e^{-j\frac{\pi\ell}{4}} + 0 + e^{-j\frac{\pi}{2}}\, e^{j\frac{\pi\ell}{4}}\right\} \tag{2-55}$$

$$f_\ell = \frac{1}{2}\left\{je^{-j\frac{\pi\ell}{4}} - je^{j\frac{\pi\ell}{4}}\right\} \equiv \sin 2\frac{\pi\ell}{8},\ \ell = 0,\ 1,\ \ldots,\ 7 \tag{2-56}$$

which is clearly the interpolated version of the original sample set.

2.4 Interpolation by the Whittaker Rule

We discussed in section 2 that by passing the sample set through an ideal low-pass filter we could reconstruct a version of the original continuous waveform. The proof of this assertion is as follows: The fundamental spectrum is obtained from Figure 2-1(c) by multiplying by equation 2-31

$$\dot{G}(f) = [G(f)*\Delta(f)]\ \frac{T}{4}\sqcap_{4/T}(f) \tag{2-57}$$

This is equivalent to the time domain convolution

$$g(t) = \int_{-\infty}^{\infty} \sum_{k=-\infty}^{\infty} g(u)\,\delta(u-kT)\ \frac{\sin\frac{4\pi}{T}(t-u)}{\frac{4\pi}{T}(t-u)}\,du \tag{2-58}$$

$$= \sum_{k=-\infty}^{\infty} g(kT)\frac{\sin\frac{4\pi}{T}(t-kT)}{\frac{4\pi}{T}(t-kT)} \tag{2-59}$$

Thus, g(t) is reconstructed as a weighted sum of sinc functions with the samples of g(t) themselves serving as the weights.

Similarly, for Figure 2-3, we obtain

$$\hat{G}(f) = [G_a(f)*W(f)]\ T_s\sqcap_{1/T_s}(f) \tag{2-60}$$

where $\hat{G}(f)$ is a corrupted version of G(f). This is equivalent to

$$\hat{g}(t) = \int_{-\infty}^{\infty} \sum_{k=0}^{N-1} g(u)\,\delta(u-kT_s)\,\frac{\sin\frac{\pi}{T_s}(t - u)}{\frac{\pi}{T_s}(t - u)}du \qquad (2\text{-}61)$$

$$\hat{g}(t) = \sum_{k=0}^{N-1} g(kT_s)\,\frac{\sin\frac{\pi}{T_s}(t - kT_s)}{\frac{\pi}{T_s}(t - kT_s)} \qquad (2\text{-}62)$$

Clearly $\hat{g}(t)$ can differ from g(t) depending on how "poorly" $\hat{G}(f)$ approximates G(f). As discussed before, this problem can be overcome by choosing $\omega(t)$ such that the convolution of W(f) and $G_a(f)$ essentially returns $G_a(f)$. This means that $G_a(f)*W(f)$ in the fundamental region ($|f| < \frac{1}{2T_s}$) is essentially the same as G(f). Then, when $G_a(f)*W(f)$ is multiplied by the ideal filter function, the resulting $\hat{G}(f)$ is also essentially the same as G(f), and equation 2-62 gives a "good" approximation to g(t).

The function expressed by equation 2-59 is formally known as the Whittaker cardinal function after E. T. Whittaker (1873 to 1956), the English scholar. Whittaker arrived at the interpolating properties of this function quite independently of the sampling theorem development used here. Equation 2-59 and its time limited version equation 2-62 are really the beginning point for the remaining work in this dissertation.

CHAPTER III

AN HISTORICAL REVIEW OF INTERPOLATING WITH THE WEIGHTED SUM OF SINC FUNCTIONS

In the first section of this chapter we briefly review polynomial interpolation and discuss the questions E. T. Whittaker asked in arriving at his cardinal interpolating function. First, we show that polynomial interpolation can be a viable scheme for fitting a continuous curve to a set of data; then, we extend the forms of polynomials to include those generated by finite difference equations. While studying the Newton difference form, Whittaker was bothered by the fact that more than one function could have the same finite difference table. He subsequently proved that the cardinal function is the lowest frequency function which passes through all the data points used to generate the differences. We also review the work of Whittaker's son, J. M. Whittaker, who extended the theory of the cardinal function by specifying conditions under which it converges. In Section 2, we discuss the digital sampling theorem from the point of view of how it relates to Whittaker's theory. Particulary important is that when the number of samples available for interpolation exceeds the product of signal bandwidth and its duration, the cardinal function and the signal which generated the samples are one and the same. Finally, in Section 3 we hint at how sample sets might be manipulated outside the region of interest in order to assure convergence of the cardinal function inside the interval.

3.1 The Whittaker Cardinal Function

The literature is replete with the theme of polynomial interpolation on data sets. The ease with which polynomials are generated and the simple form of their derivatives and integrals are no doubt fundamental reasons for the popularity. It is no surprise, then, that the cardinal function for interpolation evolved during Whittaker's study of polynomial interpolation.

To provide a brief introduction to polynomial interpolation, consider the difference y(x) - p(x), where y(x) is the given function and p(x) is an N^{th} degree polynomial to be used to approximate y(x). The central idea in interpolation is to keep this difference small. Now suppose the polynomial takes on the same values as y(x) at the tabular points $X = X_0, X_1, \ldots, X_N$. We can anticipate a result for the difference of the form [8, p. 100]

$$y(x) - p(x) = R(x - x_0)(x - x_1)\ldots(x - x_N) = R\overset{N}{\Pi}(x) \tag{3-1}$$

which is identically zero for $x = x_i$, $i = 0, 1, \ldots, N$. At any nontabular point x_j in the interval $X_0 < X_j < X_N$ and $X_j \neq X_i$, we do not expect this difference to be zero; but, if we define

$$F(x_j) = y(x_j) - p(x_j) - R\prod^{N}(x_j) \tag{3-2}$$

with

$$R = \frac{y(x_j) - p(x_j)}{\prod^{N}(x_j)} \tag{3-3}$$

we can force $F(x_j)$ to be zero. Now $F(x)$ has at least $N + 2$ zeroes. By Rolle's theorem [11, pp. 61-62][28, p. 12], $F'(x)$ is guaranteed $N + 1$ zeroes between the $N + 2$ zeroes of $F(x)$ while $F''(x)$ is guaranteed N zeroes between those of $F'(x)$. Repeatedly applying this theorem to equation 3-2, we find that $F^{(N+1)}(x)$ has at least 1 zero in the interval from x_0 to x_N, say at $x = \xi$. Then, using the fact that the "N + 1st" derivative of $p(x)$ is zero, we can write

$$F^{(N+1)}(\xi) = 0 = y^{(N+1)}(\xi) - R(N + 1)! \tag{3-4}$$

and

$$R = \frac{y^{(N+1)}(\xi)}{(N + 1)!} \tag{3-5}$$

Substituting in equation 3-1 and simplifying

$$y(x_j) - p(x_j) = \frac{y^{(N+1)}(\xi)\prod^{N}(x_j)}{(N + 1)!} \tag{3-6}$$

Since x_j is any non-tabular point and since equation 3-6 is true at the tabular points, we replace x_j with x and write

$$y(x) - p(x) = \frac{y^{(N+1)}(\xi)\prod^{N}(x)}{(N + 1)!} \tag{3-7}$$

The behavior of equation 3-7 is difficult to analyze. However, it can be shown [41, p. 123] that if $y(x)$ is an entire function, (expandible in a power series which converges for all x) then the sequence of interpolating polynomials $p_N(x)$ with $N = 3, 4, \ldots,$ defined on the interval $a \leq x \leq b$, converges uniformly to $y(x)$ on the interval. For other types of functions we can say that if some bound for $y^{(N+1)}(\xi)$ is known, then equation 3-7 may provide a useful bound on the error.

There are numerous forms for the interpolating polynomial $p(x)$. One widely used with equally spaced data is the Newton Difference Formula [28, p. 35]

$$p_K = y_0 + K\Delta y_0 + \frac{1}{2!}K^{(2)}\Delta^2 y_0 + \ldots \frac{1}{N!}K^{(N)}\Delta^N y_0 \qquad (3\text{-}8)$$

where the special notation $k^{(i)}$ is defined as

$$K^{(i)} = K(K - 1)(K - 2)\ldots(K - i + 1) \qquad (3\text{-}9)$$

and Δ^i denotes the i^{th} finite difference

$$\begin{aligned} \Delta y_0 &= y_1 - y_0 \\ \Delta^2 y_0 &= \Delta(\Delta y_0) = \Delta y_1 - \Delta y_0 = y_2 - 2y_1 - y_0 \qquad (3\text{-}10) \\ &\vdots \\ \Delta^i y_0 &= \Delta^{i-1} y_1 - \Delta^{i-1} y_0 \end{aligned}$$

Clearly equation 3-8 is true for $K = 0$; for $K = 1$,

$$P_1 = y_0 + \Delta y_0 = y_0 + y_1 - y_0 = y_1 \qquad (3\text{-}11)$$

for $K = 2$

$$p_2 = (y_0 + 2\Delta y_0) + \frac{2}{2!}\Delta^2 y_0 = (2y_1 - y_0) + y_2 - 2y_1 + y_0 = y_2 \qquad (3\text{-}12)$$

and inductively we can show the values of p_K are cotabular with y_K. We also note that K is not restricted to integer values, thus p_K is defined at nontabular points.

It was equation 3-8 that perplexed Whittaker [36, p. 181]. He noted that other functions have the same difference table as y(x)

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$ ...
⋮	⋮	⋮	⋮	⋮	⋮
$a-2\omega$	y_{-2}				
		Δy_{-2}			
$a-\omega$	y_{-1}		$\Delta^2 y_{-2}$		
		Δy_{-1}		$\Delta^3 y_{-2}$	
a	y_0		$\Delta^2 y_{-1}$		$\Delta^4 y_{-2}$ ...
		Δy_0		$\Delta^3 y_{-1}$	
$a+\omega$	y_1		$\Delta^2 y_0$		
		Δy_1			
$a+2\omega$	y_2				
⋮	⋮	⋮	⋮	⋮	⋮

(3-13)

"...for we can form a new function by adding to y(x) any analytic function which vanishes for the values a, a + ω, a − ω, ... of the argument, and this new function will have precisely the same difference table as y(x)." He called all such analytic functions "the cotabular set" and pointed out that they were all equal at the tabular points but in general not equal at nontabular points. He noted that "a-priori " there is no reason why p(x) in equation 3-8 should represent y(x) in preference to any other function in the cotabular set. Whittaker then asked two questions: (1) "Which one of the functions of the cotabular set is represented by the expansion?" (equation 3-8); (2) "Given any one function belonging to the cotabular set, is it possible to construct ...that function... which is represented by the expansion?"

In answer to his questions, Whittaker derived

$$C(x) = \sum_{n=-\infty}^{\infty} f(a + \eta\omega) \frac{\sin\frac{\pi}{\omega}(x - a - \eta\omega)}{\frac{\pi}{\omega}(x - a - \eta\omega)} \qquad (3\text{-}14)$$

as "a function which is cotabular with the given function y(x), but which has no periodic constituents of periods less than 2ω." He presented a lengthy proof which shows that C(x) is the limit of p_K given in equation 3-8 as N goes to ∞ [36, pp. 190-192]. The $f(a + \eta\omega)$, $\eta = 0, \pm 1, \ldots$, are samples from any function f(x) in the cotabular set. The fact that C(x) is generated from any one of these functions led Whittaker to call C(x) an invariant function - the simplest function belonging to the set. He defined C(x) as the cardinal function.

Professor Whittaker's son, J. M. Whittaker, made some important extensions to his father's work. Lacking in the original work was a definitive statement of under what conditions C(x) should reasonably be expected to converge. As pointed out by J. M. Whittaker [37], his father said that "when C(x) is analyzed into periodic constituents by Fourier's Integral Theorem, all constituents of periods less than 2ω are absent;" his father then proceeded to produce an example which converged but which could not be analyzed by Fourier's Theorem.

J. M. Whittaker's results are contained in his theorem: "If $\{f_n\}$ is a sequence of real numbers such that the sum of f_n^2 over all N is convergent, then the cardinal series is absolutely convergent and its sum is of the form

$$C(x) = \int_0^1 \{\phi(t)\cos\pi xt + \psi(t)\sin\pi xt\}dt \qquad (3\text{-}15)$$

where ϕ and ψ are each square integrable on [0, 1]." Here, the set $\{f_n\}$ is the same as implied in equation 3-14 except we choose a = 0 and ω = 1.

Whittaker's proof is fairly straightforward. He notes that due to the Riesz-Fisher Theorem, there are functions ϕ and ψ, and a convergent, square summable sequence $\{f_n\}$ such that

$$\int_0^1 \{\phi(t) - \phi_p(t)\}^2 dt \to 0 \qquad (3\text{-}16)$$

and

$$\int_0^1 \{\psi(t) - \psi_p(t)\}^2 dt \to 0 \qquad (3\text{-}17)$$

as $p \to \infty$ when

$$\phi_p(t) = f_0 + \sum_{n=1}^{p} (f_n + f_{-n})\cos\pi nt \tag{3-18}$$

and

$$\psi_p(t) = \sum_{n=1}^{p} (f_n + f_{-n})\sin\pi nt \tag{3-19}$$

We can show that by multiplying $\phi_p(t)$ and $\psi_p(t)$ by cosine and sine respectively and integrating on [0, 1] we have

$$\int_0^1 \phi_p(t)\cos\pi xt\,dt = f_0\frac{\sin\pi x}{\pi x} + \frac{1}{2}\sum_{n=1}^{p} (f_n+f_{-n})\left[\frac{\sin\pi(x-n)}{\pi(x-n)} + \frac{\sin\pi(x+n)}{\pi(x+n)}\right] \tag{3-20}$$

and

$$\int_0^1 \psi_p(t)\sin\pi xt\,dt = \frac{1}{2}\sum_{n=1}^{p} (f_n+f_{-n})\left[\frac{\sin\pi(x-n)}{\pi(x-n)} - \frac{\sin\pi(x+n)}{\pi(x+n)}\right] \tag{3-21}$$

Then

$$\int_0^1 \phi_p(t)\cos\pi xt\,dt + \int_0^1 \psi_p(t)\sin\pi xt\,dt = \sum_{n=-p}^{p} f_n\frac{\sin\pi(x - n)}{\pi(x - n)} \tag{3-22}$$

Equation 3-22 is the truncated version of the cardinal function. Then, form the difference between equations 3-15 and 3-22

$$\left| \int_0^1 (\phi(t)\cos\pi xt + \psi(t)\sin\pi xt)\,dt - \sum_{n=-p}^{p} f_n \frac{\sin\pi(x-n)}{\pi(x-n)} \right|$$

$$= \left| \int_0^1 \{\phi(t) - \phi_p(t)\}\cos\pi nt\,dt - \int_0^1 \{\psi(t) - \psi_p(t)\}\sin\pi nt\,dt \right|$$

$$\leq \left[\int_0^1 \{\phi(t) - \phi_p(t)\}^2 dt\right]^{1/2} + \left[\int_0^1 \{\psi(t) - \psi_p(t)\}^2 dt\right]^{1/2} \tag{3-24}$$

which approaches zero uniformly as $p \to \infty$. Thus, J. M. Whittaker proved that when a sequence of samples is square summable, equations 3-14 and 3-15 converge to the same function.

Actually, much additional work has been and is being done which extends the cardinal function to ever increasing classes of functions. One early work was by J. M. Whittaker himself [38] wherein he places some additional restrictions on sample sets of arbitrary functions to gain convergence of the cardinal series. More current work is covered by McNamee [19] where the equivalence of the communication sampling theorem and Whittaker's theory is recognized and interpreted in modern linear algebra terminology. Our purpose, though, is not to exhaustively review the cardinal function. Rather, we have shown that it is logical to pursue interpolation using the Whittaker theory.

3.2 The Sampling Theorem Approach

Interpolation with the cardinal function is not exclusively in the domain of the mathematicians. Early investigators such as Shannon [29], derive the weighted sum of sinc functions completely independently. Shannon, for instance, (page 627) while discussing certain continuous statistical functions which can be transmitted over a communication system, writes an expansion for such functions as

$$f(x) = \sum_{n=-\infty}^{\infty} f\left(\frac{n}{2W}\right) \frac{\sin\pi(2Wx-n)}{\pi(2Wx-n)} \tag{3-25}$$

He declares that "If the function f(x) is limited to the band from 0 to W cycles per second, it is completely determined by giving its ordinates at a series of discrete points spaced $\frac{1}{2W}$ seconds apart..." He notes that f(x) is represented as a sum of orthogonal functions with the samples f_n representing the coordinates in an infinite dimensional function space. Furthermore, "...if f(x) is substantially limited to a period T (i.e., $f_n = 0$, $n > N$ and $N = \frac{T}{1/2W} = 2TW$) then only 2TW coordinates are non-zero in the function space. Thus, functions limited to a band W and duration T correspond to points in a space of 2TW dimensions." Again we have Whittaker's theorem, but with a different interpretation; while Whittaker found the lowest frequency analytic function

C(x) cotabular with samples from any f(x) in the cotabular set at the tabular points, Shannon proves [30] that the cardinal function and f(x) are the same because band-limitedness and sample spacing criteria are met.

The real importance of Shannon's theorem and what makes it so distinctive from Whittaker is the information measure of "2TW." Work earlier than Shannon's by Lord Kelvin, Hartley, and Nyquist all sought or proposed similar quantifiers for the information content of signals. Modern communication theory is based on this important concept; i.e., we must send and detect - as a minimum - these 2TW coordinates of a signal space in order to reconstruct (interpolate) the original signal.

In his important paper in 1929, Hartley [14, p. 554] also came to the conclusion "that the maximum rate at which information may be transmitted over a system whose transmission is limited to frequencies lying in a restricted range, is proportional to the extent of this frequency range. From this it follows that the total amount of information which may be transmitted over such a system is proportional to the product of the frequency range which it transmits by the time during which it is available for the transmission." This conclusion was reached argumentatively by proposing various alternatives and rejecting them because they all depended on psychological considerations as opposed to physical quantities; e.g., the number of symbols available to a communications system should not be used as a measure of information because if the sender and receiver read different languages, all messages could be unintelligible.

About the same time, 1928, Nyquist [21, p. 618] also came to a similar conclusion while studying telegraph transmission theory. In an elegant hueristic argument, Nyquist proposes that we consider an arbitrary telegraph signal made up of any number and combination of dots and dashes (with a dash three times as long as a dot); the amplitude of each dot/dash is free to vary (the shape is rectangular, though) and the message, whatever its length, is assumed to be repeated indefinitely so that Fourier analysis is applicable. "The lowest frequency component has a period equal to the period of repetition...The next component is double the frequency...The third component is triple the frequency, and so forth. Certain components may be lacking...while there is always a lowest frequency, there generally is no highest..." Next, Nyquist supposes that we transmit such a signal and one identical to it except that each element of the new signal is half the duration of the original. "That is to say, everything happens twice as fast and the signals are repeated twice as frequently. It will be obvious that the analysis into sinusoidal terms (by Fourier analysis) corresponds, term for term (with the first signal) the difference being that corresponding terms are exactly twice the frequency." He next assumed that the transmission medium affected both signals linearly (deformed each the same) and noted that the second signal would be the exact counterpart of the first. "Generalizing, it may be concluded that for any given deformation of the received signal, the transmitted frequency range must be increased in direct proportion to the signaling speed, and the effect of the system at any corresponding frequencies must be the same. The conclusion is that the frequency band is directly proportional to the speed." In other words, when he doubled the speed he doubled the bandwidth but cut the duration in half. The information content of the signal was unchanged, i.e., $T \cdot W$ was constant for either signal.

The importance of the 2TW concept to interpolation is implicit in all our work in this dissertation. We assume that approximately 2TW samples are available; otherwise our interpolation schemes degenerate to the Whittaker "low frequency" cotabular algorithm (aliasing) which is abhorrent to digital signal processing. Perhaps more fundamental in our work is that we also view interpolation as a transformation or mapping of vectors in a linear vector space of 2TW dimensions. The modern approach, then, is to use matrices to describe these transformations [4, p. 107], [10, p. 32].

3.3 A Modified Sampling Theorem

Practical engineering phenomena are effectively time limited, i.e., they begin and end in finite time although the precise instants may be difficult to isolate. A theorem by Paley and Wiener known as the Paley-Wiener Condition [24] implies that such functions cannot be band-limited and thus the concept of the previous section is always violated: "A necessary and sufficient condition for square integrable function $A(\omega) \geq 0$ ($F(\omega) = A(\omega)e^{j\phi(\omega)}$) to be the Fourier spectrum of causal function ($f(t) \Longleftrightarrow F(\omega)$) is the convergence of the integral "

$$\int_{-\infty}^{\infty} \frac{|\ln A(\omega)|}{1 + \omega^2} d\omega < \infty \qquad (3\text{-}26)$$

As used by Papoulis [25, pp. 219 and 222], $F(\omega) = 0$, $\omega_1 < \omega < \omega_2$ implies that $A(\omega)$ is zero, and, therefore, $\ln A(\omega)$ is unbounded. Thus, a causal function cannot be band-limited.

This dilemma, while troublesome, is manageable. Even Shannon recognized this: "...if f(t) is 'substantially' limited...then only 2TW coordinates are nonzero..." In linear algebra terminology, if the coordinates of a function (signal) are essentially zero along all axes of an infinite dimensional function space except possibly in 2TW directions, then these 2TW coordinates adequately describe the function.

A well known trick in analyzing nonperiodic transient phenomena is to form the periodic continuation [8, p. 417] and write a Fourier series. It is also well known that the manner in which the function is extended can have serious consequences on the number of terms in the Fourier series expansion; e.g., the half cycle of a sine wave repeated indefinitely will have the classic full wave, rectified sine wave expansion containing coefficients out to ∞. However, the simple artifact of repeating the half cycle odd periodically so that a complete sine wave results, reduces the expansion to a single coefficient. As applicable to interpolation with the cardinal function the consequences are obvious - many more samples are required to interpolate the rectified sine wave than required for the pure sine wave.

In his dissertation, Campbell [5] chooses samples from finite duration functions which begin and end with first derivative discontinuities. By forming the odd periodic extension (flipping about the X and Y axis), Campbell derives a special form of the Whittaker formula which capitalizes on the reduced bandwidth of what he calls the "regionally band-limited function."

Our approach in the next chapter does not require such restrictions on the sample set. We argue that the simple periodic extension of sample sets from practical engineering systems is sufficient. Consider, for example, the single pulse consisting of one cycle of a sine wave. By the Pauley-Wiener Condition, its spectrum is infinite. Therefore, we whould not expect 2TW samples to be exactly available. The simple periodic extension of its sample set, however, reduces the problem to considering two samples. But our approach is more general; if we assume that the digital sampling and recording system is properly designed to provide "approximately" the 2TW coordinates for all input functions of interest, we then have available two different techniques for interpolation; first, the periodic extension of the sample set can be interpolated as in Chapter 5; secondly, the transient sample set itself can be interpolated as in Chapter 6.

CHAPTER IV

WHITTAKER INTERPOLATION AS A MATRIX PROCESS

This chapter begins by formulating a matrix equation which interpolates one point between every two data points in a finite length sample set. The idea was struck upon after reading Kun-Shan Lin's dissertation [17, p. 25] wherein he observed, in passing, some interesting properties of the elements in such a matrix when generated from the cardinal function. The special problem of interpolating periodic sample sets is considered first, but we then show that when the summation parameters in the matrix element generating equations are varied, the partial periodic and transient sample sets are also interpolated. When an infinitely periodic band-limited sample set is interpolated, we show that the matrix elements can be expressed in a closed form by the cotangent function. We conclude the chapter by rearranging the matrix interpolating equations into a symmetric matrix format. The special properties of these symmetric matrices are exploited in subsequent chapters.

4.1 The Whittaker Matrix for Mid-Point Reconstruction

Consider the problem of reconstructing points midway between every two samples in a periodic sample set. In particular, consider the vector of N interpolants generated by a vector of $M \cdot N$ original data points. M is an odd number of periods (windows), each N data points in length. In matrix notation we can write

$$f^N = Wx^{M \cdot N} \tag{4-1}$$

where f^N is the vector of N interpolants at intervals of $(2i - 1)T/2$, $i = 1, 2, \ldots N$; $X^{M \cdot N}$ is the vector of $M \cdot N$ original periodic data points at intervals of $(j - 1 + rN)T$, $j = 1, 2, \ldots, N$, $r = -(M - 1)/2 \ldots 0 \ldots (M - 1)/2$; W is the $N \times M \cdot N$ matrix of Whittaker coefficients

$$W = [W^{-(M-1)/2} \;\vdots\; W^0 \;\vdots\; \ldots W^{(M-1)/2}] \tag{4-2}$$

where

$$W^r = [\omega^r_{ij}] \quad i = 1, 2, \ldots, N; \; j = 1, 2, \ldots, N \tag{4-3}$$

and

$$\omega^r_{ij} = \frac{\sin\pi[i - j - rN + 1/2]}{\pi[i - j - rN + 1/2]} = \frac{2}{\pi} \frac{(-1)^{i-j-rN}}{2(i - j - rN)+1} \tag{4-4}$$

Figure 4-1 demonstrates the data format and partitioning for the case of M = 3 and a periodic sine wave. Formats for N both even and odd are shown.

Equation 4-1 can be simplified using equation 4-2 and becomes

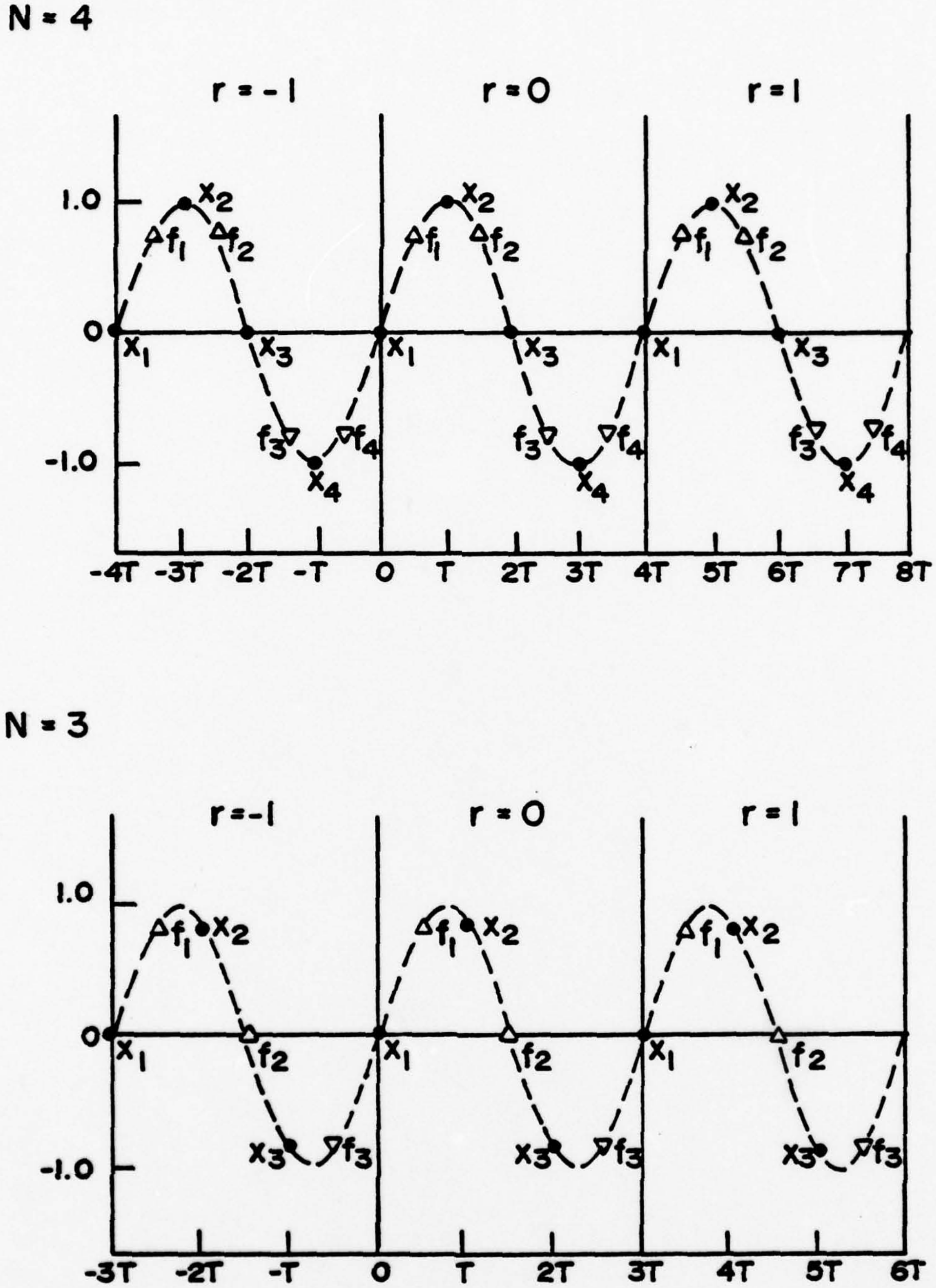


Figure 4-1

Data Format for Sine Wave

$$f^N = [\ldots W^{-1} \quad W^0 \quad W^1 \ldots] \begin{bmatrix} \vdots \\ X^N \\ X^N \\ X^N \\ \vdots \end{bmatrix} = \sum_{r=-(\frac{M-1}{2})}^{\frac{M-1}{2}} W^r X^N \tag{4-5}$$

For M = 3, N = 4, equation 4-5 yields

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \frac{2}{\pi} \begin{bmatrix} \frac{1}{9}+1-\frac{1}{7} & -\frac{1}{7}+1+\frac{1}{9} & \frac{1}{5}-\frac{1}{3}-\frac{1}{11} & -\frac{1}{3}+\frac{1}{5}+\frac{1}{13} \\ -\frac{1}{11}-\frac{1}{3}+\frac{1}{5} & \frac{1}{9}+1-\frac{1}{7} & -\frac{1}{7}+1+\frac{1}{9} & \frac{1}{5}-\frac{1}{3}-\frac{1}{11} \\ \frac{1}{13}+\frac{1}{5}-\frac{1}{3} & -\frac{1}{11}-\frac{1}{3}+\frac{1}{5} & \frac{1}{9}+1-\frac{1}{7} & -\frac{1}{7}+1+\frac{1}{9} \\ -\frac{1}{15}-\frac{1}{7}+1 & \frac{1}{13}+\frac{1}{5}-\frac{1}{3} & -\frac{1}{11}-\frac{1}{3}+\frac{1}{5} & \frac{1}{9}+1-\frac{1}{7} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} \tag{4-6}$$

and for M = 3, N = 3,

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \frac{2}{\pi} \begin{bmatrix} -\frac{1}{7}+1+\frac{1}{5} & \frac{1}{5}+1-\frac{1}{7} & -\frac{1}{3}-\frac{1}{3}+\frac{1}{9} \\ \frac{1}{9}-\frac{1}{3}-\frac{1}{3} & -\frac{1}{7}+1+\frac{1}{5} & \frac{1}{5}+1-\frac{1}{7} \\ -\frac{1}{11}+\frac{1}{5}+1 & \frac{1}{9}-\frac{1}{3}-\frac{1}{3} & -\frac{1}{7}+1+\frac{1}{5} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} \tag{4-7}$$

In general, the summed periodic form of equation 4-1 can be expressed as

$$f^N = Px^N \tag{4-8}$$

where P is the NXN matrix with

$$P = [p_{ij}] = \sum_{r=-(\frac{M-1}{2})}^{(\frac{M-1}{2})} W^r \qquad (4\text{-}9)$$

and

$$p_{ij} = \frac{2}{\pi} \sum_{r=-(\frac{M-1}{2})}^{(\frac{M-1}{2})} \frac{(-1)^{i-j-rN}}{2(i - j - rN) + 1} \qquad (4\text{-}10)$$

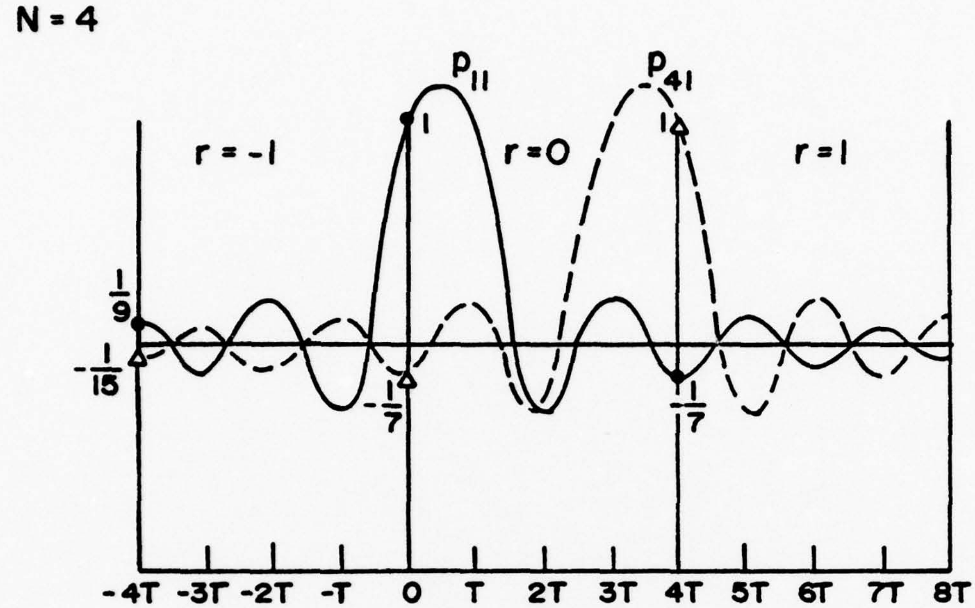
Figure 4-2 is a plot of this Whittaker kernel for p_{11} and p_{41} when M = 3 and N = 4, and for p_{11} and p_{31} when M = 3 and N = 3. Using equation 4-10, the "i^{th}" interpolant is computed as

$$f_i = \frac{2}{\pi} \sum_{j=1}^{N} \left[\sum_{r=-(\frac{M-1}{2})}^{(\frac{M-1}{2})} \frac{(-1)^{i-j-rN}}{2(i - j - rN) + 1} \right] X_j \qquad (4\text{-}11)$$

Reversing the order of summation and substitution of $j = i + p + 1 - rN$, equation 4-11 becomes

$$f_i = \frac{2}{\pi} \sum_{r=-(\frac{M-1}{2})}^{(\frac{M-1}{2})} \sum_{p=-i+rN}^{N-i-1+rN} \frac{(-1)^{i-i-p-1+rN-rN}}{2i-2(i+p+1-rN)-2rN+1} X_{i+p+1-rN}$$

(4-12)



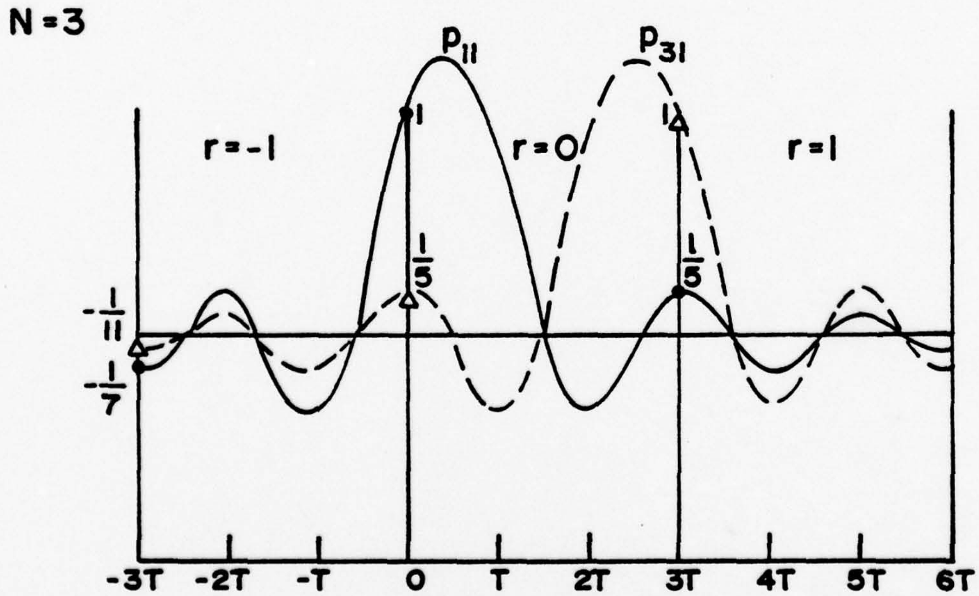


Figure 4-2

Normalized Plot of Whittaker Kernel Scale Factor = $\frac{2}{\pi}$

which reduces to

$$f_i = \frac{2}{\pi} \sum_r \sum_p \frac{(-1)^p}{2p + 1} X_{i+p+1-rN} \qquad (4\text{-}13)$$

Equation 4-11 obtains from evaluating the "i^{th}" point in equation 4-8, while equation 4-13 is the form for evaluating the same point as formulated by equation 4-1. Although the two forms are equivalent, the number of computations is not the same. For given M and N, the bracketed terms in equation 4-11 can be precomputed and stored; thus N multiplications are implied to evaluate f_i. Equation 4-13, however, requires $M \cdot N$ multiplications for the same computation.

Equations 4-11 and 4-13 can be used when M is an even number of data windows if the data can be resegmented into an odd number of windows each containing an even number of periods of periodic data samples. However, the two important cases are delineated by the present form of these equations: the case where $M = 1$, or the transient case; and the case where $M \to \infty$, *or the periodic case. Incidental* to these two important extremes are the cases where multiple cycles of some partially periodic transient event must be interpolated.

4.2 Trigonometric Representation of Matrix Elements

Equation 4-11 can be further simplified for the periodic interpolation problem. In his dissertation, Campbell

[5, Appendix B] shows that the Whittaker summation converges to a trigonometric form when the data is periodic and odd symmetric about the center of each window. A somewhat different development, assuming no special symmetry, proceeds as follows: Let $M \to \infty$, and rewrite equation 4-10 as

$$p_{ij} = \frac{(-1)^{i-j}}{\pi N} \sum_{r=-\infty}^{\infty} \frac{(-1)^{-rN}}{(\frac{i - j}{N} + \frac{1}{2N}) - r} \tag{4-14}$$

If N is restricted to an even number of data points, equation 4-14 is always positive and can be written

$$p_{ij} = \frac{(-1)^{i-j}}{\pi N} \sum_{r=-\infty}^{\infty} \frac{1}{\ell - r} \tag{4-15}$$

where

$$\ell = \frac{(i - j)}{N} + \frac{1}{2N} \tag{4-16}$$

The form of the summation in equation 4-15 can be observed by writing a few terms around $r = 0$

$$\sum \frac{1}{\ell-r} = \cdots \frac{1}{\ell+2} + \frac{1}{\ell+1} + \frac{1}{\ell} + \frac{1}{\ell-1} + \frac{1}{\ell-2} \cdots \tag{4-17}$$

Grouping like terms,

$$\sum \frac{1}{\ell-r} = \cdots (\frac{1}{\ell+2} + \frac{1}{\ell-2}) + (\frac{1}{\ell+1} + \frac{1}{\ell-1}) + \frac{1}{\ell} \tag{4-18}$$

Establishing a common denominator for each group of terms and simplifying

$$\sum \frac{1}{\ell - r} = \ldots \frac{2\ell}{\ell^2 - 2^2} + \frac{2\ell}{\ell^2 - 1^2} + \frac{1}{\ell} \tag{4-19}$$

One expansion for the cotangent of an argument [18, p. 20] is

$$\cot \pi\ell = \frac{1}{\pi\ell} + \frac{2\ell}{\pi} \sum_{k=1}^{\infty} \frac{1}{\ell^2 - k^2} \tag{4-20}$$

Using equations 4-15, 4-19, and 4-20, equation 4-14 can be simplified to

$$p_{ij} = \frac{(-1)^{i-j}}{N} \left[\frac{1}{\pi\ell} + \frac{2\ell}{\pi} \sum_{r=1}^{\infty} \frac{1}{\ell^2 - r^2} \right] \tag{4-21}$$

or,

$$p_{ij} = \frac{(-1)^{i-j}}{N} \cot \pi\ell \tag{4-22}$$

Finally, the "i^{th}" interpolant can be written as the finite length summation

$$f_i = \sum_{j=1}^{N} \left[\frac{(-1)^{i-j}}{N} \cot \pi \left(\frac{i - j}{N} + \frac{1}{2N} \right) \right] x_j \tag{4-23}$$

For the case where N is an odd number of data points, equation 4-14 alternates in sign and can be rewritten as

$$p_{ij} = \frac{(-1)^{i-j}}{\pi N} \sum_{k=-\infty}^{\infty} \left[\frac{(-1)^{-2k}}{\ell-2k} + \frac{(-1)^{-2k-1}}{\ell-2k-1} \right] \tag{4-24}$$

$$= \frac{(-1)^{i-j}}{2\pi N} \sum_{k=-\infty}^{\infty} \left[\frac{1}{\frac{\ell}{2} - k} - \frac{1}{\frac{\ell}{2} - k - \frac{1}{2}} \right]$$

where ℓ is given by equation 4-16. Proceeding as before, p_{ij} is simplified to

$$p_{ij} = \frac{(-1)^{i-j}}{2N} [\cot\pi\ell/2 - \cot\pi(\ell - 1)/2] \tag{4-25}$$

and the "i^{th}" interpolant for N odd is

$$f_i = \sum_{j=1}^{N} \frac{(-1)^{i-j}}{2N} [\cot\pi(\frac{i-j}{2N} + \frac{1}{4N}) - \cot\pi(\frac{i-j}{2N} + \frac{1}{4N} - \frac{1}{2})] X_j \tag{4-26}$$

4.3 The Symmetric Whittaker Matrix

We can rewrite equation 4-8 by reversing the sequence of interpolated points. Then,

$$g^N = Sx^N \tag{4-27}$$

where g^N is f^N written in reverse order, and S is the symmetric matrix of summed Whittaker coefficients obtained from equation 4-10 by replacing i with N + 1 - i

$$S = [s_{ij}] \; i = 1, 2, \ldots, N; \quad j = 1, 2, \ldots, N \qquad (4\text{-}28)$$

and

$$s_{ij} = \frac{2}{\pi} \sum_r \frac{(-1)^{N(1 - r) - (i + j) + 1}}{2[N(1 - r) - (i + j)] + 3} \qquad (4\text{-}29)$$

For the examples in Figure 4-1, the interpolation formulae become for N even

$$\begin{bmatrix} f_4 \\ f_3 \\ f_2 \\ f_1 \end{bmatrix} = (.616) \begin{bmatrix} .816 & -.058 & -.232 & 1.000 \\ -.058 & -.232 & 1.000 & 1.000 \\ -.232 & 1.000 & 1.000 & -.232 \\ 1.000 & 1.000 & -.232 & -.058 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} -.652 \\ -.759 \\ .759 \\ .652 \end{bmatrix} \qquad (4\text{-}30)$$

and for N odd,

$$\begin{bmatrix} f_3 \\ f_2 \\ f_1 \end{bmatrix} = (.673) \begin{bmatrix} 1.049 & -.526 & 1.000 \\ -.526 & 1.000 & 1.000 \\ 1.000 & 1.000 & -.526 \end{bmatrix} \begin{bmatrix} 0 \\ .866 \\ -.866 \end{bmatrix} = \begin{bmatrix} -.889 \\ 0 \\ .889 \end{bmatrix} \qquad (4\text{-}31)$$

The exact values are: -.707, -.707, +.707, and +.707 for N even; and -.886, 0, and .866 for N odd.

We can apply the same procedures to equations 4-22 and 4-25, yielding for the infinitely periodic case and N even

$$s_{ij} = \frac{(-1)^{i+j-1}}{N} \cot\pi\left(\frac{2N - 2i - 2j + 3}{2N}\right) \qquad (4\text{-}32)$$

and for N odd,

$$s_{ij} = \frac{(-1)^{i+j}}{2N} \left[\cot\pi\left(\frac{2N - 2i - 2j + 3}{4N}\right) - \cot\pi\left(\frac{-2i-2j+3}{4N}\right)\right]$$

(4-33)

If we expand the number of windows in Figure 4-1 to the infinitely periodic sine wave, the interpolation equation (using 4-32) becomes

$$\begin{bmatrix} f_4 \\ f_3 \\ f_2 \\ f_1 \end{bmatrix} = (.604) \begin{bmatrix} 1.000 & -.171 & -.171 & 1.000 \\ -.171 & -.171 & 1.000 & 1.000 \\ -.171 & 1.000 & 1.000 & -.171 \\ 1.000 & 1.000 & -.171 & -.171 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} -.707 \\ -.707 \\ .707 \\ .707 \end{bmatrix}$$

(4-34)

and using equation 4-33 becomes

$$\begin{bmatrix} f_3 \\ f_2 \\ f_1 \end{bmatrix} = (.667) \begin{bmatrix} 1.000 & -.500 & 1.000 \\ -.500 & 1.000 & 1.000 \\ 1.000 & 1.000 & -.500 \end{bmatrix} \begin{bmatrix} 0 \\ .866 \\ -.866 \end{bmatrix} = \begin{bmatrix} -.866 \\ 0 \\ .866 \end{bmatrix}$$

(4-35)

Figures 4-3 and 4-4 show the symmetric matrix of summed Whittaker coefficients (equations 4-29 and 4-33) for N = 9 and various choices for the number of windows;

Figures 4-5 and 4-6 show the matrix (equations 4-29 and 4-32) for N = 16. We note that the matrix elements converge rapidly as the number of windows increases.

PRINT OF WHITTAKER MATRIX
MULTIPLIER= 0.637E 00 NWIND= 1

I/J	1	2	3	4	5	6	7	8	9
1	0.059	-0.067	0.077	-0.091	0.111	-0.143	0.200	-0.333	1.000
2	-0.067	0.077	-0.091	0.111	-0.143	0.200	-0.333	1.000	1.000
3	0.077	-0.091	0.111	-0.143	0.200	-0.333	1.000	1.000	-0.333
4	-0.091	0.111	-0.143	0.200	-0.333	1.000	1.000	-0.333	0.200
5	0.111	-0.143	0.200	-0.333	1.000	1.000	-0.333	0.200	-0.143
6	-0.143	0.200	-0.333	1.000	1.000	-0.333	0.200	-0.143	0.111
7	0.200	-0.333	1.000	1.000	-0.333	0.200	-0.143	0.111	-0.091
8	-0.333	1.000	1.000	-0.333	0.200	-0.143	0.111	-0.091	0.077
9	1.000	1.000	-0.333	0.200	-0.143	0.111	-0.091	0.077	-0.067

PRINT OF WHITTAKER MATRIX
MULTIPLIER= 0.641E 00 NWIND= 3

I/J	1	2	3	4	5	6	7	8	9
1	1.024	-0.367	0.243	-0.198	0.184	-0.193	0.232	-0.350	1.000
2	-0.367	0.243	-0.198	0.184	-0.193	0.232	-0.350	1.000	1.000
3	0.243	-0.198	0.184	-0.193	0.232	-0.350	1.000	1.000	-0.350
4	-0.198	0.184	-0.193	0.232	-0.350	1.000	1.000	-0.350	0.232
5	0.184	-0.193	0.232	-0.350	1.000	1.000	-0.350	0.232	-0.193
6	-0.193	0.232	-0.350	1.000	1.000	-0.350	0.232	-0.193	0.184
7	0.232	-0.350	1.000	1.000	-0.350	0.232	-0.193	0.184	-0.198
8	-0.350	1.000	1.000	-0.350	0.232	-0.193	0.184	-0.198	0.243
9	1.000	1.000	-0.350	0.232	-0.193	0.184	-0.198	0.243	-0.367

4-3 Symmetric Whittaker Matrix
Number of Windows = 1 and 3, N = 9

PRINT OF WHITTAKER MATRIX
MULTIPLIER= 0.640E 00 NWIND= 9

I/J	1	2	3	4	5	6	7	8	9
1	0.998	-0.345	0.225	-0.183	0.172	-0.184	0.226	-0.347	1.000
2	-0.345	0.225	-0.183	0.172	-0.184	0.226	-0.347	1.000	1.000
3	0.225	-0.183	0.172	-0.184	0.226	-0.347	1.000	1.000	-0.347
4	-0.183	0.172	-0.184	0.226	-0.347	1.000	1.000	-0.347	0.226
5	0.172	-0.184	0.226	-0.347	1.000	1.000	-0.347	0.226	-0.184
6	-0.184	0.226	-0.347	1.000	1.000	-0.347	0.226	-0.184	0.172
7	0.226	-0.347	1.000	1.000	-0.347	0.226	-0.184	0.172	-0.183
8	-0.347	1.000	1.000	-0.347	0.226	-0.184	0.172	-0.183	0.225
9	1.000	1.000	-0.347	0.226	-0.184	0.172	-0.183	0.225	-0.345

PRINT OF WHITTAKER MATRIX
MULTIPLIER= 0.640E 00 NWIND=999

I/J	1	2	3	4	5	6	7	8	9
1	1.000	-0.347	0.227	-0.185	0.174	-0.185	0.227	-0.347	1.000
2	-0.347	0.227	-0.185	0.174	-0.185	0.227	-0.347	1.000	1.000
3	0.227	-0.185	0.174	-0.185	0.227	-0.347	1.000	1.000	-0.347
4	-0.185	0.174	-0.185	0.227	-0.347	1.000	1.000	-0.347	0.227
5	0.174	-0.185	0.227	-0.347	1.000	1.000	-0.347	0.227	-0.185
6	-0.185	0.227	-0.347	1.000	1.000	-0.347	0.227	-0.185	0.174
7	0.227	-0.347	1.000	1.000	-0.347	0.227	-0.185	0.174	-0.185
8	-0.347	1.000	1.000	-0.347	0.227	-0.185	0.174	-0.185	0.227
9	1.000	1.000	-0.347	0.227	-0.185	0.174	-0.185	0.227	-0.347

4-4 Symmetric Whittaker Matrix,
Number of Windows = 9 and 999, N = 9

PRINT OF WHITTAKER MATRIX
MULTIPLIER= 0.637E 00 NWIND= 1

I/J	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	-0.032	0.034	-0.037	0.040	-0.043	0.048	-0.053	0.059	-0.067	0.077	-0.091	0.111	-0.143	0.200	-0.333	1.000
2	0.034	-0.037	0.040	-0.043	0.048	-0.053	0.059	-0.067	0.077	-0.091	0.111	-0.143	0.200	-0.333	1.000	1.000
3	-0.037	0.040	-0.043	0.048	-0.053	0.059	-0.067	0.077	-0.091	0.111	-0.143	0.200	-0.333	1.000	1.000	-0.333
4	0.040	-0.043	0.048	-0.053	0.059	-0.067	0.077	-0.091	0.111	-0.143	0.200	-0.333	1.000	1.000	-0.333	0.200
5	-0.043	0.048	-0.053	0.059	-0.067	0.077	-0.091	0.111	-0.143	0.200	-0.333	1.000	1.000	-0.333	0.200	-0.143
6	0.048	-0.053	0.059	-0.067	0.077	-0.091	0.111	-0.143	0.200	-0.333	1.000	1.000	-0.333	0.200	-0.143	0.111
7	-0.053	0.059	-0.067	0.077	-0.091	0.111	-0.143	0.200	-0.333	1.000	1.000	-0.333	0.200	-0.143	0.111	-0.091
8	0.059	-0.067	0.077	-0.091	0.111	-0.143	0.200	-0.333	1.000	1.000	-0.333	0.200	-0.143	0.111	-0.091	0.077
9	-0.067	0.077	-0.091	0.111	-0.143	0.200	-0.333	1.000	1.000	-0.333	0.200	-0.143	0.111	-0.091	0.077	-0.067
10	0.077	-0.091	0.111	-0.143	0.200	-0.333	1.000	1.000	-0.333	0.200	-0.143	0.111	-0.091	0.077	-0.067	0.059
11	-0.091	0.111	-0.143	0.200	-0.333	1.000	1.000	-0.333	0.200	-0.143	0.111	-0.091	0.077	-0.067	0.059	-0.053
12	0.111	-0.143	0.200	-0.333	1.000	1.000	-0.333	0.200	-0.143	0.111	-0.091	0.077	-0.067	0.059	-0.053	0.048
13	-0.143	0.200	-0.333	1.000	1.000	-0.333	0.200	-0.143	0.111	-0.091	0.077	-0.067	0.059	-0.053	0.048	-0.043
14	0.200	-0.333	1.000	1.000	-0.333	0.200	-0.143	0.111	-0.091	0.077	-0.067	0.059	-0.053	0.048	-0.043	0.040
15	-0.333	1.000	1.000	-0.333	0.200	-0.143	0.111	-0.091	0.077	-0.067	0.059	-0.053	0.048	-0.043	0.040	-0.037
16	1.000	1.000	-0.333	0.200	-0.143	0.111	-0.091	0.077	-0.067	0.059	-0.053	0.048	-0.043	0.040	-0.037	0.034

PRINT OF WHITTAKER MATRIX
MULTIPLIER= 0.635E 00 NWIND= 3

I/J	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.954	-0.283	0.146	-0.085	0.050	-0.024	0.005	0.013	-0.029	0.047	-0.067	0.092	-0.129	0.190	-0.328	1.000
2	-0.283	0.146	-0.085	0.050	-0.024	0.005	0.013	-0.029	0.047	-0.067	0.092	-0.129	0.190	-0.328	1.000	1.000
3	0.146	-0.085	0.050	-0.024	0.005	0.013	-0.029	0.047	-0.067	0.092	-0.129	0.190	-0.328	1.000	1.000	-0.328
4	-0.085	0.050	-0.024	0.005	0.013	-0.029	0.047	-0.067	0.092	-0.129	0.190	-0.328	1.000	1.000	-0.328	0.190
5	0.050	-0.024	0.005	0.013	-0.029	0.047	-0.067	0.092	-0.129	0.190	-0.328	1.000	1.000	-0.328	0.190	-0.129
6	-0.024	0.005	0.013	-0.029	0.047	-0.067	0.092	-0.129	0.190	-0.328	1.000	1.000	-0.328	0.190	-0.129	0.092
7	0.005	0.013	-0.029	0.047	-0.067	0.092	-0.129	0.190	-0.328	1.000	1.000	-0.328	0.190	-0.129	0.092	-0.067
8	0.013	-0.029	0.047	-0.067	0.092	-0.129	0.190	-0.328	1.000	1.000	-0.328	0.190	-0.129	0.092	-0.067	0.047
9	-0.029	0.047	-0.067	0.092	-0.129	0.190	-0.328	1.000	1.000	-0.328	0.190	-0.129	0.092	-0.067	0.047	-0.029
10	0.047	-0.067	0.092	-0.129	0.190	-0.328	1.000	1.000	-0.328	0.190	-0.129	0.092	-0.067	0.047	-0.029	0.013
11	-0.067	0.092	-0.129	0.190	-0.328	1.000	1.000	-0.328	0.190	-0.129	0.092	-0.067	0.047	-0.029	0.013	0.005
12	0.092	-0.129	0.190	-0.328	1.000	1.000	-0.328	0.190	-0.129	0.092	-0.067	0.047	-0.029	0.013	0.005	-0.024
13	-0.129	0.190	-0.328	1.000	1.000	-0.328	0.190	-0.129	0.092	-0.067	0.047	-0.029	0.013	0.005	-0.024	0.050
14	0.190	-0.328	1.000	1.000	-0.328	0.190	-0.129	0.092	-0.067	0.047	-0.029	0.013	0.005	-0.024	0.050	-0.085
15	-0.328	1.000	1.000	-0.328	0.190	-0.129	0.092	-0.067	0.047	-0.029	0.013	0.005	-0.024	0.050	-0.085	0.146
16	1.000	1.000	-0.328	0.190	-0.129	0.092	-0.067	0.047	-0.029	0.013	0.005	-0.024	0.050	-0.085	0.146	-0.283

4-5 Symmetric Whittaker Matrix,
Number of Windows = 1 and 3, N = 16

PRINT OF WHITTAKER MATRIX
MULTIPLIER= 0.635E 00 NWIND= 9

I/J	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.986	-0.312	0.172	-0.109	0.071	-0.043	0.022	-0.002	-0.016	0.036	-0.057	0.085	-0.123	0.186	-0.326	1.000
2	-0.312	0.172	-0.109	0.071	-0.043	0.022	-0.002	-0.016	0.036	-0.057	0.085	-0.123	0.186	-0.326	1.000	1.000
3	0.172	-0.109	0.071	-0.043	0.022	-0.002	-0.016	0.036	-0.057	0.085	-0.123	0.186	-0.326	1.000	1.000	-0.326
4	-0.109	0.071	-0.043	0.022	-0.002	-0.016	0.036	-0.057	0.085	-0.123	0.186	-0.326	1.000	1.000	-0.326	0.186
5	0.071	-0.043	0.022	-0.002	-0.016	0.036	-0.057	0.085	-0.123	0.186	-0.326	1.000	1.000	-0.326	0.186	-0.123
6	-0.043	0.022	-0.002	-0.016	0.036	-0.057	0.085	-0.123	0.186	-0.326	1.000	1.000	-0.326	0.186	-0.123	0.085
7	0.022	-0.002	-0.016	0.036	-0.057	0.085	-0.123	0.186	-0.326	1.000	1.000	-0.326	0.186	-0.123	0.085	-0.057
8	-0.002	-0.016	0.036	-0.057	0.085	-0.123	0.186	-0.326	1.000	1.000	-0.326	0.186	-0.123	0.085	-0.057	0.036
9	-0.016	0.036	-0.057	0.085	-0.123	0.186	-0.326	1.000	1.000	-0.326	0.186	-0.123	0.085	-0.057	0.036	-0.016
10	0.036	-0.057	0.085	-0.123	0.186	-0.326	1.000	1.000	-0.326	0.186	-0.123	0.085	-0.057	0.036	-0.016	-0.002
11	-0.057	0.085	-0.123	0.186	-0.326	1.000	1.000	-0.326	0.186	-0.123	0.085	-0.057	0.036	-0.016	-0.002	0.022
12	0.085	-0.123	0.186	-0.326	1.000	1.000	-0.326	0.186	-0.123	0.085	-0.057	0.036	-0.016	-0.002	0.022	-0.043
13	-0.123	0.186	-0.326	1.000	1.000	-0.326	0.186	-0.123	0.085	-0.057	0.036	-0.016	-0.002	0.022	-0.043	0.071
14	0.186	-0.326	1.000	1.000	-0.326	0.186	-0.123	0.085	-0.057	0.036	-0.016	-0.002	0.022	-0.043	0.071	-0.109
15	-0.326	1.000	1.000	-0.326	0.186	-0.123	0.085	-0.057	0.036	-0.016	-0.002	0.022	-0.043	0.071	-0.109	0.172
16	1.000	1.000	-0.326	0.186	-0.123	0.085	-0.057	0.036	-0.016	-0.002	0.022	-0.043	0.071	-0.109	0.172	-0.312

PRINT OF WHITTAKER MATRIX
MULTIPLIER= 0.635E 00 NWIND=999

I/J	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1.000	-0.325	0.184	-0.120	0.081	-0.053	0.030	-0.010	-0.010	0.030	-0.053	0.081	-0.120	0.184	-0.325	1.000
2	-0.325	0.184	-0.120	0.081	-0.053	0.030	-0.010	-0.010	0.030	-0.053	0.081	-0.120	0.184	-0.325	1.000	1.000
3	0.184	-0.120	0.081	-0.053	0.030	-0.010	-0.010	0.030	-0.053	0.081	-0.120	0.184	-0.325	1.000	1.000	-0.325
4	-0.120	0.081	-0.053	0.030	-0.010	-0.010	0.030	-0.053	0.081	-0.120	0.184	-0.325	1.000	1.000	-0.325	0.184
5	0.081	-0.053	0.030	-0.010	-0.010	0.030	-0.053	0.081	-0.120	0.184	-0.325	1.000	1.000	-0.325	0.184	-0.120
6	-0.053	0.030	-0.010	-0.010	0.030	-0.053	0.081	-0.120	0.184	-0.325	1.000	1.000	-0.325	0.184	-0.120	0.081
7	0.030	-0.010	-0.010	0.030	-0.053	0.081	-0.120	0.184	-0.325	1.000	1.000	-0.325	0.184	-0.120	0.081	-0.053
8	-0.010	-0.010	0.030	-0.053	0.081	-0.120	0.184	-0.325	1.000	1.000	-0.325	0.184	-0.120	0.081	-0.053	0.030
9	-0.010	0.030	-0.053	0.081	-0.120	0.184	-0.325	1.000	1.000	-0.325	0.184	-0.120	0.081	-0.053	0.030	-0.010
10	0.030	-0.053	0.081	-0.120	0.184	-0.325	1.000	1.000	-0.325	0.184	-0.120	0.081	-0.053	0.030	-0.010	-0.010
11	-0.053	0.081	-0.120	0.184	-0.325	1.000	1.000	-0.325	0.184	-0.120	0.081	-0.053	0.030	-0.010	-0.010	0.030
12	0.081	-0.120	0.184	-0.325	1.000	1.000	-0.325	0.184	-0.120	0.081	-0.053	0.030	-0.010	-0.010	0.030	-0.053
13	-0.120	0.184	-0.325	1.000	1.000	-0.325	0.184	-0.120	0.081	-0.053	0.030	-0.010	-0.010	0.030	-0.053	0.081
14	0.184	-0.325	1.000	1.000	-0.325	0.184	-0.120	0.081	-0.053	0.030	-0.010	-0.010	0.030	-0.053	0.081	-0.120
15	-0.325	1.000	1.000	-0.325	0.184	-0.120	0.081	-0.053	0.030	-0.010	-0.010	0.030	-0.053	0.081	-0.120	0.184
16	1.000	1.000	-0.325	0.184	-0.120	0.081	-0.053	0.030	-0.010	-0.010	0.030	-0.053	0.081	-0.120	0.184	-0.325

4-6 Symmetric Whittaker Matrix,
Number of Windows = 9 and 999, N = 16

CHAPTER V

CYCLICAL DECOMPSITION OF THE PERIODIC WHITTAKER MATRIX

We begin this chapter by decomposing the periodic Whittaker matrix into the product of 4 special matrices whose operations on the data vector x can be directly implemented by the Cooley-Tukey [3], [34] factorization algorithm (FFT). The first section develops these matrices from eigenvalue-eigenvector considerations and an equivalence transformation. In Section 2 we rewrite the periodic Whittaker matrix in a special cyclic form. Finite difference equations are then written which express the eigenvalue-eigenvector equations in discrete form. In Section 3 we solve these difference equations and obtain "nice" analytical expressions for the eigenvalues and eigenvectors. A brief description of the algorithm for implementing interpolation via the cyclical decomposition is presented in Section 4.

The overall purpose of this chapter is to show that the matrix interpolating equation 4-27 can be implemented with fewer than the implied N^2 operation. Implicit in the chapter is the new interpretation of the workings of an ideal digital filter; i.e., although we never invoke the sampling theorem or discrete filter theory, we arrive at an equivalent algorithm.

5.1 Equivalence Transform

The eigenvalues λ_i and eigenvectors v^i of equation 4-8 are found from the matrix equation

$$Pv^i = \lambda_i v^i, \; v^i \neq 0, \; i = 0, 1, \ldots, N-1 \qquad (5\text{-}1)$$

The diagonal matrix Λ of all eigenvalues and the corresponding matrix $V = [v^0 \mid v^1 \mid \cdots \mid v^{N-1}]$ of all eigenvectors also solve equation 5-1

$$PV = V\Lambda \qquad (5\text{-}2)$$

Now the symmetric Whittaker matrix equation 4-27 was developed from P by premultiplying with a permutation matrix T which interchanges the first and last rows, second and next-to-last rows, and so on

$$TPV = TV\Lambda \qquad (5\text{-}3)$$

and with

$$S = TP \qquad (5\text{-}4)$$

we have

$$SV = TV\Lambda \qquad (5\text{-}5)$$

where the elements of S are given by equations 4-32 and 4-33. If V is non-singular, we can rewrite equation 5-5 as

$$S = TV\Lambda V^{-1} \qquad (5\text{-}6)$$

and the symmetric interpolating equation 4-27 becomes

$$g = TV\Lambda V^{-1}x \tag{5-7}$$

Equation 5-6 is recognizable as an equivalence transformation; i.e., S and Λ are equivalent [39, p. 18]. Our objective in this chapter is to show that elements of V and V^{-1} are simply samples from the Fourier kernel, and that the λ_i are related to the discrete Fourier Transform of elements of S; the entire equation 5-7 can be implemented as a DFT of x followed by N multiplications by the λ_i, followed by another DFT. To accomplish this, we first develop a "cyclical" representation for the matrix elements of S, and following the approach of Grey [13, p. 17], we solve the cyclical eigenvector equation for Λ and V.

5.2 The Cyclical Whittaker Matrix

Equations 4-32 and 4-33 can be rewritten using the substitution

$$\ell = i + j - 2, \quad 0 \leq i + j - 2 \leq N - 1 \tag{5-8}$$

Then, N even

$$s_\ell = \frac{(-1)^{\ell+1}}{N} \cot\pi\left(\frac{2N - 1 - 2\ell}{2N}\right) \tag{5-9}$$

N odd

$$s_\ell = \frac{(-1)^{\ell}}{2N}\left[\cot\pi\left(\frac{2N - 1 - 2\ell}{4N}\right) - \cot\pi\left(\frac{-1 - 2\ell}{4N}\right)\right] \tag{5-10}$$

A little trig reduces equations 5-9 and 5-10 for N even

$$s_\ell = \frac{(-1)^\ell}{N} \cot\pi\left(\frac{1 + 2\ell}{2N}\right) \tag{5-11}$$

and for N odd

$$s_\ell = \frac{(-1)^\ell}{N} \csc\pi\left(\frac{1 + 2\ell}{2N}\right) \tag{5-12}$$

The corresponding S matrix becomes

$$S = [s_\ell] = \begin{bmatrix} s_0 & s_1 & s_2 & \cdots & s_{N-2} & s_{N-1} \\ s_1 & s_2 & s_3 & \cdots & s_{N-1} & s_0 \\ s_2 & s_3 & s_4 & \cdots & s_0 & s_1 \\ \vdots & & & & & \vdots \\ s_{N-1} & s_0 & s_1 & \cdots & s_{N-3} & s_{N-2} \end{bmatrix} \tag{5-13}$$

where

$$s_\ell = s_{N-\ell-1}, \quad \ell = 0, 1, \ldots, N - 1 \tag{5-14}$$

For example, for N = 3 or 4, we have

$$\begin{bmatrix} s_0 & s_1 & s_2 \\ s_1 & s_2 & s_0 \\ s_2 & s_0 & s_1 \end{bmatrix} = \begin{bmatrix} s_0 & s_1 & s_0 \\ s_1 & s_0 & s_0 \\ s_0 & s_0 & s_1 \end{bmatrix} \tag{5-15}$$

and

$$
\begin{bmatrix} s_0 & s_1 & s_2 & s_3 \\ s_1 & s_2 & s_3 & s_0 \\ s_2 & s_3 & s_0 & s_1 \\ s_3 & s_0 & s_1 & s_2 \end{bmatrix} = \begin{bmatrix} s_0 & s_1 & s_1 & s_0 \\ s_1 & s_1 & s_0 & s_0 \\ s_1 & s_0 & s_0 & s_1 \\ s_0 & s_0 & s_1 & s_1 \end{bmatrix} \qquad (5\text{-}16)
$$

The matrix equation 5-13 is of a cyclic form [13], [15], and [16]. The system of simultaneous equations implied by equation 5-5 with elements as shown in equation 5-13 can be written out for a general vector v and constant λ.

$$
\begin{aligned}
s_0v_0+s_1v_1+s_2v_2 + \cdots\ s_{N-2}v_{N-2}+s_{N-1}v_{N-1} &= \lambda v_{N-1} \\
s_1v_0+s_2v_1+s_3v_2 + \cdots\ s_{N-1}v_{N-2}+s_0v_{N-1} &= \lambda v_{N-2} \\
\vdots \qquad\qquad\qquad\qquad \vdots \qquad\quad \vdots \qquad &\quad \vdots \\
s_{N-1}v_0+s_0v_1+s_1v_2+\cdots\ s_{N-3}v_{N-2}+s_{N-2}v_{N-1} &= \lambda v_0
\end{aligned}
\qquad (5\text{-}17)
$$

which is equivalent to the N difference equations, $m = 0, 1, \ldots, N - 1$,

$$
\sum_{k=m}^{N-1} s_K v_{K-m} + \sum_{k=0}^{m-1} s_K v_{N+K-m} = \lambda v_{N-m-1} \qquad (5\text{-}18)
$$

5.3 Eigenvalues and Eigenvectors of the Cyclic Form

One way of finding the λ and v for equation 5-18 is to assume forms for v and show that the resulting expressions work; we assume for the elements of v

$$v_K = \rho^K \tag{5-19}$$

Substituting in equation 5-18 and simplifying

$$\rho^{-m}\sum_{k=m}^{N-1} s_K \rho^K + \rho^N \rho^{-m}\sum_{k=0}^{m-1} s_K \rho^K = \lambda \rho^N \rho^{-m} \rho^{-1} \tag{5-20}$$

If we further assume

$$\rho^N = 1 \tag{5-21}$$

we have

$$\lambda = \sum_{k=0}^{N-1} s_K \rho^{K+1} \tag{5-22}$$

and the general eigenvector (normalized) can be written as

$$v = \frac{1}{\sqrt{N}}(1,\ \rho^1,\ \rho^2,\ \ldots,\ \rho^{N-1})^T \tag{5-23}$$

Now one function which has the properties in equation 5-19 and 5-21 is [1], [12]

$$\rho^K = e^{-j2\frac{\pi K}{N}} \tag{5-24}$$

Then we can write for the N different eigenvalues and eigen-vectors

$$\lambda_\ell = \sum_{k=0}^{N-1} s_K e^{-j2\frac{\pi(K+1)\ell}{N}},\quad \ell = 0,\ 1,\ \ldots,\ N-1 \tag{5-25}$$

$$v^\ell = \frac{1}{\sqrt{N}}(1,\ e^{-j2\frac{\pi\ell}{N}},\ e^{-j4\frac{\pi\ell}{N}},\ \ldots,e^{-j2\frac{(N-1)\pi\ell}{N}})^T,\ \ell=0,1,\ldots,N-1 \tag{5-26}$$

To find the matrix of all eigenvectors we use equation 5-26 and write

$$V = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & e^{-j2\frac{\pi}{N}} & e^{-j4\frac{\pi}{N}} & \dots & e^{-j2\frac{(N-1)\pi}{N}} \\ 1 & e^{-j4\frac{\pi}{N}} & e^{-j8\frac{\pi}{N}} & \dots & e^{-j4\frac{(N-1)\pi}{N}} \\ \vdots & & & & \\ 1 & e^{-j2\frac{(N-1)\pi}{N}} & e^{-j4\frac{(N-1)\pi}{N}} & \dots & e^{-j2\frac{(N-1)^2\pi}{N}} \end{bmatrix} \tag{5-27}$$

which is a unitary matrix; i.e., the inner product of any two columns is

$$\langle v^p, v^q \rangle = \frac{1}{N}\sum_{K=0}^{N-1} e^{-j2\frac{\pi kp}{N}} e^{+j2\frac{\pi kq}{N}} \tag{5-28}$$

$$= \frac{1}{N}\sum_{K=0}^{N-1} e^{-j2\frac{\pi(p-q)K}{N}} = \begin{Bmatrix} 1, & p = q \\ 0, & p \neq q \end{Bmatrix} \tag{5-29}$$

Therefore, we can write V^{-1} as V^*, and

$$V^* = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{+j2\frac{\pi}{N}} & e^{+j4\frac{\pi}{N}} & \cdots & e^{+j2\frac{(N-1)\pi}{N}} \\ 1 & e^{+j4\frac{\pi}{N}} & e^{+j8\frac{\pi}{N}} & \cdots & e^{+j4\frac{(N-1)\pi}{N}} \\ \vdots & & & & \\ 1 & e^{+j2\frac{(N-1)\pi}{N}} & e^{+j4\frac{(N-1)\pi}{N}} & \cdots & e^{+j2\frac{(N-1)^2\pi}{N}} \end{bmatrix} \tag{5-30}$$

Finally,

$$S = TV\Lambda V^* \tag{5-31}$$

where Λ is the diagonal matrix of eigenvalues of P given by equation 5-25.

We can verify that equation 5-31 is true by computing the "m, n^{th}" element of S. First, expand S as

$$S = [s_\ell] = \frac{1}{N}\sum_{\ell=0}^{N-1} \lambda_\ell \hat{v}^\ell v^{\ell *} \tag{5-32}$$

with

$$\hat{v}^\ell = Tv^\ell \tag{5-33}$$

Then for the m, n^{th} element we write, using equations 5-8 and 5-25

$$s_{m,n} = s_{m+n-2} = \frac{1}{N}\sum_{\ell=0}^{N-1} e^{-j2\frac{\pi(N-m)\ell}{N}} e^{j2\frac{\pi(n-1)\ell}{N}} \lambda_\ell \tag{5-34}$$

$$= \frac{1}{N}\sum_{\ell=0}^{N-1} e^{j2\frac{\pi(m + n - 1)\ell}{N}} \sum_{K=0}^{N-1} s_K e^{-j2\frac{\pi(k + 1)\ell}{N}} \qquad (5\text{-}35)$$

$$= \frac{1}{N}\sum_{K=0}^{N-1} s_k \sum_{\ell=0}^{N-1} e^{-j2\frac{\pi(k - m - n + 2)\ell}{N}} \qquad (5\text{-}36)$$

$$= s_{m + n - 2}$$

Some further simplification for the λ_ℓ can be made. When N is even, equation 5-25 can be rewritten

$$\lambda_\ell = e^{-j2\frac{\pi\ell}{N}} \sum_{K=0}^{\frac{N}{2}-1} s_K e^{-j2\frac{\pi k\ell}{N}} + e^{-j2\frac{\pi\ell}{N}} \sum_{k=\frac{N}{2}}^{N-1} s_k e^{-j2\frac{\pi k\ell}{N}} \qquad (5\text{-}38)$$

let p=N - k - 1 in the second summation, and using equation 5-14

$$\lambda_\ell = e^{-j2\frac{\pi\ell}{N}} \sum_{K=0}^{\frac{N}{2}-1} s_k e^{-j2\frac{\pi k\ell}{N}} + e^{-j2\frac{\pi\ell}{N}} \sum_{p=\frac{N}{2}-1}^{0} s_p e^{j2\frac{\pi(p+1)\ell}{N}} \qquad (5\text{-}39)$$

$$= e^{-j\frac{\pi\ell}{N}} \left\{ \sum_{K=0}^{\frac{N}{2}-1} s_K \left(e^{-j\frac{\pi(1 + 2k)\ell}{N}} + e^{+j\frac{\pi(1 + 2k)\ell}{N}}\right) \right\} \qquad (5\text{-}40)$$

$$= e^{-j\frac{\pi\ell}{N}} \sum_{K=0}^{\frac{N}{2}-1} 2s_K \cos\frac{\pi}{N}(1 + 2k)\ell \qquad (5\text{-}41)$$

When N is odd, equation 5-25 becomes

$$\lambda_\ell = e^{-j2\frac{\pi\ell}{N}}\left\{\sum_{K=0}^{\frac{N-3}{2}} s_\ell e^{-j2\frac{\pi k\ell}{N}} + \sum_{p=0}^{\frac{N-3}{2}} s_p e^{j2\frac{\pi(p+1)\ell}{N}} + s_{\frac{N-1}{2}} e^{-j2\frac{\pi(N-1)\ell}{2N}}\right\} \quad (5\text{-}42)$$

$$= e^{-j\frac{\pi\ell}{N}}\left\{\sum_{K=0}^{\frac{N-3}{2}} 2s_k \cos\frac{\pi(1 + 2k)\ell}{N} + (-1)^\ell s_{\frac{N-1}{2}}\right\} \quad (5\text{-}43)$$

Substituting equations 5-11 and 5-12 for S_ℓ, we finally have for N even

$$\lambda_\ell = e^{-j\frac{\pi\ell}{N}} \sum_{K=0}^{\frac{N}{2}-1} \frac{2}{N}(-1)^k \cot\pi\left(\frac{1 + 2k}{2N}\right)\cos\pi\left(\frac{1 + 2k}{N}\right)\ell \quad (5\text{-}44)$$

Equation 5-44 reveals that whenever $\ell = \frac{N}{2}$, $\lambda_\ell = 0$ which means that even ordered periodic Whittaker matrix is always singular. When N is odd, we find

$$\lambda_\ell = e^{-j\frac{\pi\ell}{N}}\left\{\sum_{K=0}^{\frac{N-3}{2}} \frac{2}{N}(-1)^k \csc\pi\left(\frac{1+2k}{2N}\right)\cos\pi\left(\frac{1+2k}{N}\right)\ell + \frac{(-1)^{2\frac{\ell+N-1}{2}}}{N}\right\} \quad (5\text{-}45)$$

5.4 The FFT Algorithm

In summary, we write equation 5-7 as

$$g = (TV)\Lambda(V^*x) \quad (5\text{-}46)$$

We see that the product V*x (using 5-30) is simply the Discrete Fourier Transform (DFT) of the data sequence x. Furthermore, if X = V*x, we can write

$$F = \Lambda X \tag{5-47}$$

and

$$g = (TV)F \tag{5-48}$$

which is simply an N point multiplication by the diagonal matrix of eigenvalues with elements given by equations 5-44 and 5-45, followed by another DFT of the product. The DFT's are implementable, of course, with the Cooley-Tukey Fast Fourier Transform algorithm [34]. Subroutine "FFTINT" in the Appendix implements this algorithm.

CHAPTER VI

DECOMPOSITION OF THE REAL SYMMETRIC WHITTAKER MATRIX

The cyclical decomposition of Chapter V only applies when the Whittaker matrix S is periodic. When S is formulated with the transient generating equations in Chapter IV, S is still symmetric but has lost its cyclical properties. This chapter briefly reviews the orthogonal similarity transformation for real symmetric matrices as a prelude to introducing the Eigen-Filter, Cross-Correlation Algorithm which implements transient interpolation. In Section 1 we prove that the real symmetric matrix S is orthogonally similar to a diagonal matrix, and that similarity implies the diagonal matrix is the matrix of eigenvalues of S. In Section 2 we discuss subroutine SYMEIG [20] in the Appendix which finds the eigenvalues and eigenvectors of S using the Francis QR Algorithm. In Section 3 we show that the similarity transformation on S can be viewed as a cross-correlation process wherein we first measure the similarity of the eigenvectors of S to the data vector x before the interpolation process is begun. By only using significant correlants, the number of operations necessary to implement $g = Sx$ can be significantly reduced.

6.1 Orthogonal Transformation

We know that vectors in a linear vector space of $N = 2TW$ dimensions can be expressed as linear combinations of basis vectors [10, p. 75]. We can write arbitrary vectors x and g with respect to the basis

$$Q = [q^1 \mid q^2 \mid \cdots \mid q^N] \qquad (6\text{-}1)$$

as

$$x = a_1 q^1 + a_2 q^2 + \ldots, a_N q^N \qquad (6\text{-}2)$$

and

$$g = b_1 q^1 + b_2 q^2 + \ldots, b_N q^N \qquad (6\text{-}3)$$

or, in matrix notation

$$x = Qa \qquad (6\text{-}4)$$

$$g = Qb \qquad (6\text{-}5)$$

Now, a linear transformation S which maps x to g is written as

$$g = Sx \qquad (6\text{-}6)$$

Substituting equations 6-4 and 6-5

$$Qb = SQa \qquad (6\text{-}7)$$

$$b = Q^{-1}SQa \qquad (6\text{-}8)$$

Now define

$$T = Q^{-1}SQ \tag{6-9}$$

and, after solving for S, equation 6-6 becomes

$$g = QTQ^{-1}x \tag{6-10}$$

Equation 6-9 is a similarity transformation with the important property that T and S have the same eigenvalues; in other words

$$\det(T - \gamma I) = \det(S - \gamma I) \tag{6-11}$$

When S is a symmetric matrix, we can write its eigenvalue-eigenvector equation as

$$Sq^i = \gamma_i q^i \tag{6-12}$$

where γ_i, i = 1, 2, ..., N, are the eigenvalues of S and the q^i are chosen as the associated eigenvectors. Using inner product notation, we find

$$\gamma_i \langle q^i, q^i \rangle = \langle \gamma_i q^i, q^i \rangle = \langle S \quad q^i, q^i \rangle \tag{6-13}$$

or

$$\gamma_i = \frac{\langle Sq^i, q^i \rangle}{\langle q^i, q^i \rangle} \tag{6-14}$$

By definition of inner product, the denominator in equation 6-14 is real. The numerator is real because it equals its own conjugate. Therefore, the γ_i are real as the quotient of two real numbers is real. Now, owing to a theorem by

Schur [7, p. 106], if any matrix S has only real eigenvalues, then it is orthogonally similar to an upper triangular matrix; that is,

$$T = Q^T S Q \tag{6-15}$$

where T is upper triangular with diagonal elements t_i equal to the γ_i of S and with $Q^TQ = QQ^T = I$. This can be shown by hypothesizing an eigenvector q^1 for the eigenvalue t_1 and forming the complete orthonormal set Q by Gram Schmit (or Householder as discussed later). Then we have

$$Q_1^T S Q_1 = \begin{bmatrix} q^{1T} \\ q^{2T} \\ \vdots \\ q^{NT} \end{bmatrix} [Sq^1 \mid Sq^2 \mid \cdots \mid Sq^N] = \begin{bmatrix} q^{1T} \\ q^{2T} \\ \vdots \\ q^{NT} \end{bmatrix} [t_1 q^1 \mid Sq^2 \mid \cdots \mid Sq^N] \tag{6-16}$$

$$= \left[\begin{array}{c|c} t_1 q^{1T} q^1 & q^{1T} Sq^2 \cdots q^{1T} Sq^N \\ \hline t_1 q^{2T} q^1 & \\ & S_2 \\ t_1 q^{NT} q^1 & \end{array}\right] = \begin{bmatrix} t_1 & * \\ 0 & S_2 \end{bmatrix} \tag{6-17}$$

(The symbol "*" is used to indicate that the corresponding matrix elements are not germaine to the discussion.)

We can repeat the process on S_2 by finding an orthonormal set Q_2 such that

$$\begin{bmatrix} 1 & 0 \\ 0 & Q_2^T \end{bmatrix} [Q_1^T S Q_1] \begin{bmatrix} 1 & 0 \\ 0 & Q_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & Q_2^T \end{bmatrix} \begin{bmatrix} t_1 & * \\ 0 & S_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & Q_2 \end{bmatrix}$$

$$= \begin{bmatrix} t_1 & * \\ 0 & Q_2^T S_2 Q_2 \end{bmatrix} = \left[\begin{array}{cc|c} t_1 & * & * \\ 0 & t_2 & \\ \hline \multicolumn{2}{c|}{0} & S_3 \end{array}\right] \qquad (6\text{-}18)$$

After N - 1 repetitions we can write

$$Q = Q_1 \begin{bmatrix} 1 & 0 \\ 0 & Q_2 \end{bmatrix} \begin{bmatrix} I_2 & 0 \\ 0 & Q_3 \end{bmatrix} \cdots \begin{bmatrix} I_{N-2} & 0 \\ 0 & Q_{N-1} \end{bmatrix} \qquad (6\text{-}19)$$

and

$$T = Q^T S Q = \begin{bmatrix} t_1 & & * \\ & t_2 \ddots & \\ 0 & & t_N \end{bmatrix} \qquad (6\text{-}20)$$

But when S is symmetric, $Q^T SQ$ is symmetric. Therefore, T must be diagonal. Thus

$$S = Q \Gamma Q^T \qquad (6\text{-}21)$$

where Γ is the diagonal matrix of all eigenvalues of S. Equation 6-21 is now an Orthogonal Similarity Transformation. Finally, we can write the symmetric Whittaker interpolating equation as

$$g = Q\Gamma Q^T x \tag{6-22}$$

6.2 Francis QR Algorithm

Eigenvalues and eigenvectors for the symmetric Whittaker matrix can in general be computed from some numerical technique. The method adopted in our work is to make this computation using the subroutine SYMEIG in the Appendix. This program was developed at UNM by Dr. Cleve Moler [20, Chapter 7] and is based on the Francis QR transformation [9] (Q implies orthogonal matrix and R is a right triangular matrix). Dr. Moler's algorithm is the "Real-Symmetric" adaptation of the more general technique, and consists of two fundamental steps: (1) reduction of a real-symmetric matrix S to a tridiagonal matrix using Householder transformations; (2) reduction of the tridiagonal matrix to a diagonal matrix using the iterative Francis QR transformation. The first step is required because the QR Algorithm would be too expensive, in terms of computer time, to use on a general NxN matrix S.

The first step of the algorithm (S to tridiagonal) can be described in terms of matrix products; however, the computer code is quite different owing to the fact that the matrix operations simplify due to symmetries (see lines 0010 to 0068 in SYMEIG, Appendix). Given the symmetric matrix

$$S = \begin{bmatrix} s_0 & s_1 & s_2 & \cdots & s_{N-2} & s_{N-1} \\ s_1 & s_2 & s_3 & \cdots & s_{N-1} & s_0 \\ \vdots & & & & & \\ s_{N-1} & \cdot & \cdot & \cdots & s_{N-3} & s_{N-2} \end{bmatrix} \tag{6-23}$$

define

$$\sigma_1 = \mathrm{SIGN}(s_1)\sqrt{s_1^2 + s_2^2 + \ldots s_{N-1}^2} \tag{6-24}$$

Let

$$u' = \begin{bmatrix} 0 \\ s_1 + \sigma_1 \\ s_2 \\ \vdots \\ s_{N-1} \end{bmatrix} \tag{6-25}$$

and

$$\pi_1 = \sigma_1(s_1 + \sigma_1) = \tfrac{1}{2}\langle u', u' \rangle \tag{6-26}$$

Now let

$$P_1 = I - \frac{1}{\pi_1} u'u'^T = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & p_{22} & \cdots & p_{2N} \\ \vdots & \vdots & & \vdots \\ 0 & p_{N2} & \cdots & p_{NN} \end{bmatrix} \tag{6-27}$$

Equation 6-27 is a Householder reflection with the special properties

$$P_1 = P_1^T \tag{6-28}$$

and

$$P_1^T P_1 = P_1 P_1^T = I \tag{6-29}$$

that is, P_1 is orthogonal. From section 6.1 we know

$$S_2 = P_1^T S P_1 \tag{6-30}$$

is a similarity transformation and that S_2 and S have the same eigenvalues.

The special feature of the Householder transformation as generated by equations 6-24 to 6-27 and as implemented by equation 6-30 is the introduction of zeroes in the first row and column of S

$$S_2 = \left[\begin{array}{c|cccc} s_0 & -\sigma_1 & 0 & \dots & 0 \\ \hline -\sigma_1 & & & & \\ 0 & & & & \\ \vdots & & \hat{S}_2 & & \\ 0 & & & & \end{array}\right] \tag{6-31}$$

Now, form a P_2 which introduces zeroes in the first row and first column of $\hat{S}_2$

$$
S_3 = \left[\begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & \hat{P}_2^T & \\ 0 & & & \end{array}\right] \left[\begin{array}{c|ccc} s_0 & -\sigma_1 & \cdots & 0 \\ \hline -\sigma_1 & & & \\ 0 & & \hat{S}_2 & \\ \vdots & & & \\ 0 & & & \end{array}\right] \left[\begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & \hat{P}_2 & \\ 0 & & & \end{array}\right] \tag{6-32}
$$

Let

$$
\omega = \begin{bmatrix} -\sigma_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{6-33}
$$

Then

$$
S_3 = P_2^T S_2 P_2 = \begin{bmatrix} 1 & 0 \\ 0 & \hat{P}_2^T \end{bmatrix} \begin{bmatrix} s_o & \omega^T \\ \omega & \hat{S}_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \hat{P}_2 \end{bmatrix} \tag{6-34}
$$

and

$$
S_3 = \begin{bmatrix} 1 & 0 \\ 0 & \hat{P}_2^T \end{bmatrix} \begin{bmatrix} s_0 & \omega^T \hat{P}_2 \\ \omega & \hat{S}_2 \hat{P}_2 \end{bmatrix} = \left[\begin{array}{c|c} s_0 & \omega^T \hat{P}_2 \\ \hline \hat{P}_2^T \omega & \hat{P}_2^T \hat{S}_2 \hat{P}_2 \end{array}\right] \tag{6-35}
$$

But

$$
\omega^T \hat{P}_2 = [-\sigma_1 \quad 0 \ \ldots \ 0] \tag{6-36}
$$

and

AD-A061 030 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO F/G 9/3
DIGITAL SIGNAL INTERPOLATION USING MATRIX TECHNIQUES AND THE WH--ETC(U)
MAY 78 J C WHEELER
UNCLASSIFIED AFIT-CI-78-120D NL

2 OF 3
AD
A061030

$$\hat{P}_2^T \omega = \begin{bmatrix} -\sigma_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{6-37}$$

Then

$$S_3 = \left[\begin{array}{c|c} s_0 & -\sigma_1 \quad 0 \quad \cdots \\ \hline -\sigma_1 & \\ 0 & \hat{P}_2^T \hat{S}_2 \hat{P}_2 \\ \vdots & \\ 0 & \end{array}\right] = \left[\begin{array}{cc|cccc} s_0 & -\sigma_1 & 0 & \cdots & & 0 \\ -\sigma_1 & s_2^{(1)} & -\sigma_2 & 0 & \cdots & 0 \\ \hline 0 & -\sigma_2 & & & & \\ \vdots & & & \hat{S}_3 & & \\ 0 & 0 & & & & \end{array}\right] \tag{6-38}$$

after a total of N - 2 applications of Householder

$$P_{N-2}^T \cdots P_1^T S P_1 \cdots P_{N-2}$$

$$= \begin{bmatrix} s_0 & -\sigma_1 & & & & 0 \\ -\sigma_1 & s_2^{(1)} & -\sigma_2 & & & \\ & -\sigma_2 & s_4^{(2)} & -\sigma_3 & & \\ & & -\sigma_3 & \ddots & & \\ & & & & s_{N-4}^{(N-2)} & s_{N-3}^{(N-2)} \\ & 0 & & & s_{N-3}^{(N-2)} & s_{N-2}^{(N-2)} \end{bmatrix} = S_{N-2} \tag{6-39}$$

which is the desired tridiagonal form.

The second major step of SYMEIG (tridiagonal to diagonal via Francis QR) operates iteratively on equation 6-39. As with the first step, we describe the algorithm with matrix products although the actual implementation is different (see lines 0069 to 0110 in SYMEIG). The basic idea is to let $A_1 = S_{N-2}$ where S_{N-2} is as given in equation 6-39; then, factor A_1 into the product of an orthogonal matrix Q_1 and a right triangular matrix R_1. Next, reverse the products and form $A_2 = R_1Q_1$ and then factor again. Continuing, we write the following sequence

$$A_1 = Q_1R_1 \tag{6-40}$$

let

$$A_2 = R_1Q_1 = Q_2R_2 \tag{6-41}$$

and

$$A_3 = R_2Q_2 = Q_3R_3 \tag{6-42}$$

$$\vdots$$

and

$$A_k = R_{k-1}Q_{k-1} = Q_kR_k \tag{6-43}$$

and finally,

$$A_{k+1} = R_kQ_k \tag{6-44}$$

Now, solving for R_k in equation 6-43 and substituting in equation 6-44

$$A_{k+1} = Q_k^TA_kQ_k \tag{6-45}$$

In other words, equation 6-45 is again a similarity transformation. The theory of the QR Algorithm is that as $k \to \infty$ the off diagonal elements of A_{k+1} tend to zero; i.e., A_{k+1} tends to diagonal form when A_1 is tridiagonal. The factorizations to $Q_k R_k$ are actually implemented by Householder transformations just as we used in going to tridiagonal form; that is, we first zero below diagonal elements in A_k by N - 2 applications of Householder transformations

$$P_{N-2}^T \dots P_1^T A_k = R_k \tag{6-46}$$

then form

$$Q_k = (P_{N-2}^T \dots P_1^T)^T \tag{6-47}$$

and since

$$A_k = Q_k R_k \tag{6-48}$$

we have

$$A_{k+1} = R_k Q_k = P_{N-2}^T \dots P_1^T A_k P_1 \dots P_{N-2} \tag{6-49}$$

The iterations are necessary because even though R_k has all zeroes below the diagonal, when $R_k Q_k$ is formed to complete the similarity transform, A_{k+1} again becomes tridiagonal. In the final form, A_{k+1} is diagonal to working precision on the machine. The products of all Q_i and P_i are orthogonal so we conclude with the orthogonal similarity transformation

$$\Gamma = A_{k+1} = Q^T S Q \tag{6-50}$$

where Γ is the diagonal matrix of eigenvalues and Q is the accumulated products of Householder transformations. We note that Q is also the matrix of eigenvectors.

As described, the QR algorithm may converge slowly or not at all. However, it can be shown [40], that by modifying A_1 to $\hat{A}_1 = A_1 - \sigma I$ where σ is a root of the lower 2 x 2 submatrix of A_1, and then decomposing $\hat{A}_1$ as described by equations 6-40 to 6-45, that convergence is always assured. Experience has shown that this simple "shift" generally produces convergence at the average rate of only one or two iterations per eigenvalue. This shift modification is implemented in lines 0083 to 0087 in subroutine SYMEIG. Figures 6-1 through 6-4 show the results of applying SYMEIG to the Whittaker matrices shown in Figures 4-1 through 4-4.

6.3 The Eigenfilter Cross-Correlation Algorithm

If we solve equation 4-29 (element generating equation for the transient symmetric Whittaker matrix) for $s_{1,N}$, $s_{i,N-i+1}$, and $s_{i,N-i+2}$ for i = 2, ..., N, we find these minor diagonal and minor subdiagonal elements all equal to $2/\pi$. Then, the symmetric matrix interpolating equation can be modified to

$$g = Sx = (S - D)x + Dx = \hat{S}x + Dx \tag{6-51}$$

PRINT OF EIGENVALUES.
MULTIPLIER= 0.157E 01 NWIND= 1

I/J	1	2	3	4	5	6	7	8	9
1	0.421	-0.894	0.990	-0.999	-1.000	-1.000	1.000	1.000	1.000

PRINT OF EIGENVECTORS.
MULTIPLIER= 0.737E 00 NWIND= 1

I/J	1	2	3	4	5	6	7	8	9
1	1.000	0.756	-0.453	-0.230	0.037	-0.013	0.101	0.007	-0.018
2	-0.482	-0.023	-0.512	-0.763	0.463	-0.106	0.689	0.199	-0.160
3	0.363	-0.138	0.534	0.340	0.814	0.145	0.258	0.723	-0.138
4	-0.309	0.222	-0.461	0.028	0.017	0.853	-0.467	0.586	0.443
5	0.280	-0.285	0.358	-0.301	-0.163	0.343	0.572	-0.222	0.972
6	-0.266	0.347	-0.228	0.477	0.278	-0.823	0.009	0.139	0.773
7	0.262	-0.422	0.051	-0.528	-0.509	-0.529	-0.328	0.812	0.098
8	-0.271	0.531	0.228	0.335	-0.754	0.074	0.706	0.454	-0.211
9	0.303	-0.735	-0.802	0.619	-0.182	0.056	0.375	0.056	-0.074

PRINT OF EIGENVALUES.
MULTIPLIER= 0.162E 01 NWIND= 3

I/J	1	2	3	4	5	6	7	8	9
1	-0.990	1.000	-0.961	0.959	-0.961	-0.961	0.961	0.961	0.961

PRINT OF EIGENVECTORS.
MULTIPLIER= 0.672E 00 NWIND= 3

I/J	1	2	3	4	5	6	7	8	9
1	0.534	0.916	0.324	-0.965	0.058	-0.017	-0.143	0.126	0.121
2	-0.074	-0.692	0.783	-0.204	0.560	-0.108	-0.657	0.496	0.268
3	-0.126	0.584	-0.430	0.439	0.860	0.206	-0.533	0.323	-0.557
4	0.265	-0.491	0.014	-0.451	-0.025	0.935	0.332	0.253	-0.806
5	-0.379	0.401	0.306	0.353	-0.176	0.361	0.594	1.000	0.422
6	0.481	-0.305	-0.504	-0.176	0.315	-0.877	0.546	0.640	-0.168
7	-0.580	0.194	0.528	-0.065	-0.496	-0.615	-0.086	0.150	-0.950
8	0.684	-0.057	-0.225	0.373	-0.848	0.050	-0.756	0.509	-0.020
9	-0.812	-0.140	-0.796	-0.739	-0.247	0.065	-0.372	0.304	0.243

6-1 Eigenvalues and Eigenvectors of Symmetric Matrix, Number of Windows = 1 and 3, N = 9

PRINT OF EIGFNVALUES.
MULTIPLIER= 0.153E 01 NWIND= 9

I/J	1	2	3	4	5	6	7	8	9
1	0.995	-0.996	1.000	-1.000	-1.000	-1.000	1.000	1.000	1.000

PRINT OF EIGENVECTORS.
MULTIPLIER= 0.783E 00 NWIND= 9

I/J	1	2	3	4	5	6	7	8	9
1	0.751	0.455	0.854	0.288	-0.059	0.026	0.207	0.033	-0.012
2	-0.594	-0.085	0.133	0.649	-0.506	0.116	0.743	-0.028	0.101
3	0.517	-0.094	-0.355	-0.392	-0.723	-0.200	0.258	-0.378	0.551
4	-0.442	0.227	0.390	0.055	0.028	-0.801	-0.390	-0.004	0.665
5	0.361	-0.335	-0.298	0.270	0.143	-0.297	0.214	1.000	0.269
6	-0.270	0.428	0.138	-0.459	-0.259	0.732	-0.122	0.564	0.458
7	0.164	-0.511	0.077	0.429	0.404	0.550	-0.236	-0.358	0.709
8	-0.033	0.589	-0.338	-0.146	0.736	-0.043	0.688	-0.199	0.301
9	-0.144	-0.673	0.621	-0.694	0.236	-0.082	0.477	0.037	0.008

PRINT OF EIGENVALUES.
MULTIPLIER= 0.156E 01 NWIND=999

I/J	1	2	3	4	5	6	7	8	9
1	-1.000	-1.000	-1.000	-1.000	1.000	1.000	1.000	1.000	1.000

PRINT OF EIGENVECTORS.
MULTIPLIER= 0.898E 00 NWIND=999

I/J	1	2	3	4	5	6	7	8	9
1	-0.410	0.204	0.116	-0.013	1.000	-0.103	0.037	0.058	0.040
2	0.077	0.623	0.353	-0.109	-0.154	-0.379	0.464	-0.473	-0.322
3	0.284	0.377	-0.387	0.423	0.176	0.267	-0.544	-0.472	-0.261
4	-0.187	-0.121	0.258	0.643	-0.128	-0.602	-0.308	0.324	-0.364
5	0.218	-0.113	0.391	0.100	0.135	0.636	0.249	0.300	-0.665
6	-0.254	0.229	-0.714	-0.361	-0.091	-0.139	0.075	0.394	-0.543
7	0.274	-0.128	0.340	-0.697	0.104	-0.221	-0.653	-0.039	-0.245
8	-0.619	-0.554	-0.030	-0.058	-0.077	0.098	0.016	-0.644	-0.342
9	0.617	-0.516	-0.326	0.073	0.355	-0.413	0.373	-0.159	-0.145

6-2 Eigenvalues and Eigenvectors of Symmetric Matrix, Number of Windows = 9 and 999, N = 9

PRINT OF EIGENVALUES.
MULTIPLIER= 0.157E 01 NWIND= 1

I/J	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	-0.385	-0.980	0.857	-1.000	0.998	-1.000	-1.000	-1.000	-1.000	-1.000	1.000	1.000	1.000	1.000	1.000	1.000

PRINT OF EIGENVECTORS.
MULTIPLIER= 0.823E 00 NWIND= 1

I/J	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.851	0.446	-0.676	-0.141	-0.262	-0.025	-0.022	-0.001	0.005	-0.001	0.071	-0.001	-0.009	0.001	-0.004	-0.030
2	-0.408	0.292	0.102	-0.616	-0.551	-0.308	-0.252	0.032	0.121	-0.014	0.563	0.047	-0.089	0.116	-0.116	-0.010
3	0.303	-0.354	0.020	0.158	0.406	-0.451	-0.263	0.300	0.418	-0.089	0.295	0.182	-0.062	0.509	-0.330	-0.175
4	-0.252	0.346	-0.075	0.119	-0.243	0.119	0.362	0.594	0.273	0.112	-0.346	0.014	0.351	0.363	0.014	-0.502
5	0.223	-0.322	0.109	-0.279	0.109	-0.053	0.138	0.105	-0.113	0.712	0.367	-0.025	0.660	-0.228	0.179	-0.112
6	-0.203	0.292	-0.134	0.314	0.001	-0.028	-0.136	-0.431	0.443	0.406	-0.097	0.120	0.400	0.218	-0.236	0.615
7	0.190	-0.261	0.154	-0.318	-0.091	0.630	-0.183	0.019	0.415	-0.079	0.005	-0.362	-0.032	0.474	0.464	0.239
8	-0.180	0.228	-0.173	0.273	0.165	0.166	-0.673	0.321	-0.336	0.218	0.139	0.429	-0.078	0.092	0.546	0.002
9	0.174	-0.192	0.192	-0.199	-0.226	-0.012	0.316	-0.138	0.187	-0.121	-0.183	1.000	-0.024	0.006	0.260	0.100
10	-0.170	0.152	-0.212	0.099	0.275	-0.476	0.524	-0.279	0.052	-0.073	0.277	-0.186	-0.108	0.279	0.710	0.007
11	0.169	-0.106	0.235	0.027	-0.309	-0.394	0.054	0.295	-0.635	-0.040	-0.264	-0.129	0.159	0.480	0.002	0.540
12	-0.169	0.049	-0.263	-0.153	0.325	0.218	0.118	0.268	-0.076	-0.715	0.301	0.122	0.593	-0.121	-0.089	0.339
13	0.173	0.025	0.298	0.320	-0.311	-0.169	-0.395	-0.456	-0.007	-0.433	-0.013	-0.084	0.567	0.001	0.229	-0.443
14	-0.181	-0.132	-0.347	-0.423	0.240	0.183	-0.050	-0.501	-0.443	0.065	-0.192	0.146	0.108	0.568	-0.242	-0.359
15	0.195	0.307	0.423	0.376	-0.036	0.476	0.356	-0.125	-0.270	0.048	0.626	0.122	-0.117	0.302	-0.247	-0.036
16	-0.227	-0.672	-0.569	0.434	-0.589	0.123	0.105	-0.003	-0.035	0.003	0.287	0.007	-0.039	0.024	-0.032	-0.000

PRINT OF EIGENVALUES.
MULTIPLIER= 0.158E 01 NWIND= 3

I/J	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	-0.019	-0.996	1.000	-0.994	0.994	-0.994	-0.994	-0.994	-0.994	-0.994	0.994	0.994	0.994	0.994	0.994	0.994

PRINT OF EIGENVECTORS.
MULTIPLIER= 0.775E 00 NWIND= 3

I/J	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	-0.449	-0.448	1.000	0.151	0.442	0.057	-0.005	-0.025	0.032	0.001	-0.124	-0.134	-0.014	0.026	-0.070	-0.035
2	0.359	-0.237	-0.381	0.573	0.386	0.416	0.034	-0.163	0.310	0.096	-0.369	-0.484	-0.106	-0.035	-0.255	-0.323
3	-0.330	0.328	0.182	-0.255	-0.456	0.364	0.178	0.067	0.449	0.456	0.213	-0.181	-0.287	-0.162	-0.065	-0.611
4	0.314	-0.331	-0.072	-0.050	0.369	-0.166	-0.183	0.610	-0.017	0.485	0.446	0.114	-0.520	0.322	0.222	-0.103
5	-0.304	0.309	-0.001	0.228	-0.226	0.205	-0.624	0.357	0.106	-0.289	-0.522	-0.035	-0.544	0.221	0.204	0.249
6	0.297	-0.279	0.053	-0.354	0.096	0.021	0.176	0.253	0.504	-0.520	-0.293	0.289	-0.009	-0.458	0.541	-0.265
7	-0.293	0.245	-0.094	0.438	0.056	-0.640	0.181	0.071	0.396	0.033	0.075	-0.379	0.349	0.277	0.589	-0.129
8	0.291	-0.210	0.127	-0.323	-0.174	-0.215	-0.597	-0.515	0.348	0.162	0.184	-0.541	-0.106	-0.252	0.171	0.375
9	-0.290	0.171	-0.137	0.235	0.272	0.040	0.294	0.251	-0.147	-0.061	0.335	-0.273	-0.328	-0.847	0.058	0.455
10	0.291	-0.127	0.184	-0.134	-0.337	0.499	0.341	0.322	-0.392	-0.177	0.075	-0.666	0.186	0.301	0.378	0.166
11	-0.293	0.075	-0.210	-0.114	0.373	0.420	-0.432	-0.286	-0.461	0.253	-0.003	0.083	0.269	-0.174	0.662	-0.328
12	0.298	-0.009	0.235	0.266	-0.336	-0.298	0.324	-0.214	-0.376	0.557	-0.586	0.149	-0.331	-0.243	0.332	0.014
13	-0.305	-0.080	-0.259	-0.376	0.268	0.069	0.521	-0.573	0.084	-0.141	-0.049	-0.012	-0.592	0.466	0.204	0.206
14	0.316	0.210	0.280	0.442	-0.104	-0.045	-0.099	-0.387	-0.237	-0.581	0.542	0.077	-0.402	0.015	0.129	-0.451
15	-0.332	-0.416	-0.289	-0.279	-0.167	-0.513	-0.115	0.137	-0.461	-0.253	-0.199	-0.442	-0.188	-0.137	-0.228	-0.530
16	0.361	0.800	0.259	-0.449	0.576	-0.214	0.003	0.093	-0.137	-0.021	-0.273	-0.313	-0.046	0.022	-0.164	-0.134

6-3 Eigenvalues and Eigenvectors of Symmetric Matrix, Number of Windows = 1 and 3, N = 16

PRINT OF EIGENVALUES.
MULTIPLIER= 0.158E 01 NWIND= 9

I/J	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	-0.005	-0.999	-0.999	-0.999	1.000	-0.999	-0.999	-0.999	-0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999

PRINT OF EIGENVECTORS.
MULTIPLIER= 0.747E 00 NWIND= 9

I/J	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.378	0.460	0.221	-0.058	1.000	-0.004	0.001	-0.007	0.001	0.365	0.336	0.356	-0.107	0.043	-0.001	-0.000
2	-0.348	0.145	0.807	0.063	-0.464	-0.007	0.155	-0.179	-0.078	0.256	0.281	0.405	-0.131	0.161	0.605	-0.029
3	0.338	-0.289	-0.036	0.450	0.272	-0.056	0.482	-0.522	0.052	-0.429	-0.271	-0.062	-0.208	0.239	0.420	-0.517
4	-0.333	0.331	-0.203	-0.228	-0.145	-0.439	0.135	-0.291	0.574	-0.279	0.105	0.429	-0.509	0.122	-0.509	-0.163
5	0.329	-0.336	0.303	0.268	0.054	-0.470	-0.411	0.233	0.361	-0.183	-0.310	0.169	-0.212	0.301	0.112	0.723
6	-0.327	0.321	-0.096	0.618	0.021	0.390	0.040	0.287	0.382	0.458	-0.172	-0.406	-0.244	0.576	-0.108	-0.057
7	0.325	-0.289	0.181	-0.377	-0.082	0.162	0.554	0.440	0.357	-0.037	0.481	-0.459	-0.528	-0.174	0.065	0.099
8	-0.324	0.246	-0.151	0.287	0.134	-0.582	0.387	0.431	-0.436	0.104	-0.351	-0.035	-0.393	-0.549	0.203	0.096
9	0.324	-0.193	0.110	-0.205	-0.178	0.279	-0.139	-0.169	0.240	0.640	-0.708	0.262	-0.352	-0.448	-0.097	-0.261
10	-0.324	0.129	-0.057	0.111	0.215	0.360	-0.545	-0.514	0.120	-0.265	0.168	-0.341	-0.476	-0.509	0.328	0.310
11	0.325	-0.052	-0.038	0.014	-0.245	-0.507	-0.374	-0.329	-0.534	0.403	0.314	-0.456	-0.444	0.294	-0.210	-0.188
12	-0.327	-0.040	-0.063	-0.761	0.267	0.058	0.291	-0.296	-0.248	0.114	-0.475	-0.211	-0.119	0.529	0.164	0.435
13	0.329	0.150	-0.202	0.247	-0.278	0.590	0.227	-0.014	-0.565	-0.241	0.038	0.481	-0.421	0.134	-0.288	0.441
14	-0.333	-0.290	0.388	-0.181	0.278	0.209	-0.426	0.509	-0.337	-0.419	-0.136	0.107	-0.391	0.193	-0.182	-0.558
15	0.338	0.486	-0.538	-0.340	-0.250	0.006	-0.341	0.369	0.089	-0.131	-0.054	0.124	-0.124	0.223	0.746	-0.242
16	-0.348	-0.779	-0.627	0.094	0.168	0.010	-0.037	0.054	0.022	0.412	0.400	0.462	-0.132	0.090	0.249	0.023

PRINT OF EIGENVALUES.
MULTIPLIER= 0.158E 01 NWIND=999

I/J	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	-0.000	-1.000	-1.000	1.000	-1.000	-1.000	-1.000	-1.000	-1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

PRINT OF EIGENVECTORS.
MULTIPLIER= 0.791E 00 NWIND=999

I/J	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	-0.316	-0.128	0.276	1.000	0.032	0.029	-0.216	0.317	-0.019	0.432	0.167	-0.013	-0.010	-0.038	0.170	-0.103
2	0.316	0.180	-0.193	0.261	0.106	0.078	-0.314	0.683	-0.062	-0.399	-0.451	0.081	0.091	0.058	-0.545	0.310
3	-0.316	0.712	-0.122	0.145	-0.218	0.256	0.051	-0.008	0.263	-0.327	0.234	-0.399	-0.443	0.377	0.194	0.268
4	0.316	-0.002	0.516	0.131	-0.102	0.426	0.484	0.199	-0.051	-0.515	-0.012	0.383	-0.187	0.013	0.204	-0.507
5	-0.316	-0.419	-0.531	0.088	-0.142	0.279	0.328	0.231	0.193	-0.334	-0.192	-0.356	0.392	-0.409	0.375	0.005
6	0.316	-0.103	0.198	-0.027	-0.038	-0.257	-0.034	0.080	0.807	0.204	-0.480	-0.443	-0.392	-0.111	-0.011	-0.293
7	-0.316	0.190	-0.041	-0.046	0.054	-0.612	0.525	0.307	-0.156	0.153	-0.526	0.362	-0.136	0.211	0.387	0.202
8	0.316	-0.099	0.068	0.019	-0.904	-0.163	-0.197	0.037	-0.122	0.046	-0.091	-0.137	0.443	0.547	0.238	-0.045
9	-0.316	0.070	-0.073	-0.023	0.452	0.025	0.164	0.015	0.008	-0.004	-0.111	-0.292	0.412	0.675	-0.312	-0.659
10	0.316	-0.050	0.093	0.018	0.550	0.421	-0.208	-0.201	0.287	0.101	-0.237	0.165	0.279	0.369	0.610	0.403
11	-0.316	-0.131	-0.207	-0.082	-0.223	0.547	-0.342	-0.229	-0.360	0.212	-0.658	0.123	-0.479	0.067	-0.006	-0.235
12	0.316	0.332	0.240	0.057	0.234	-0.063	0.044	-0.076	-0.707	0.007	-0.238	-0.728	0.055	-0.310	0.271	-0.034
13	-0.316	0.312	0.140	0.090	0.019	-0.389	-0.596	-0.325	0.156	-0.566	-0.207	0.254	0.264	-0.290	0.240	-0.310
14	0.316	-0.425	-0.542	0.175	0.251	-0.378	-0.216	0.012	-0.218	-0.341	0.272	-0.042	-0.511	0.300	0.313	-0.226
15	-0.316	-0.607	0.559	0.109	0.028	-0.134	0.087	-0.308	-0.084	-0.457	-0.173	-0.258	-0.133	0.242	-0.254	0.480
16	0.316	0.166	-0.369	0.647	-0.097	-0.064	0.441	-0.733	0.066	0.006	-0.230	0.144	0.088	-0.033	-0.285	0.041

6-4 Eigenvalues and Ei vectors of Symmetric
Matrix, Number of dows = 9 and 999, N = 16

where

$$D = \frac{2}{\pi}\begin{bmatrix} & & & & 1 \\ & & & 1 & 1 \\ & 0 & & & \\ & & \cdot^{\cdot^{\cdot}} 1 & 1 & \\ 1 & 1 & & 0 & \end{bmatrix} \tag{6-52}$$

Now Dx is a simple linear interpolating scheme and can be implemented as a scaled sum of adjacent elements in x. Also, $\hat{S}$ remains symmetric and therefore remains orthogonally similar to a diagonal matrix of eigenvalues; then

$$g = \hat{Q}\hat{\Gamma}\hat{Q}^T x + Dx \tag{6-53}$$

The similarity transformation can be implemented as

$$\hat{g} = \hat{Q}\hat{\Gamma}\hat{Q}^T x = \sum_{j=1}^{N} \hat{\gamma}_j \hat{q}^j \langle \hat{q}^j, x\rangle \tag{6-54}$$

where the inner product notation inside the summation indicates that $\langle q^j, x\rangle$ is a scalar quantity. If we let

$$\rho_j = \langle \hat{q}^j, x\rangle \tag{6-55}$$

equation 6-54 is implemented as

$$\begin{bmatrix} \hat{g}_1 \\ \hat{g}_2 \\ \vdots \\ \hat{g}_N \end{bmatrix} = \hat{\gamma}_1 \rho_1 \begin{bmatrix} \hat{q}_1^1 \\ \cdot \\ \cdot \\ \cdot \\ \hat{q}_N^1 \end{bmatrix} + \hat{\gamma}_2 \rho_2 \begin{bmatrix} \hat{q}_1^2 \\ \cdot \\ \cdot \\ \cdot \\ \hat{q}_N^2 \end{bmatrix} + \dots \hat{\gamma}_N \rho_N \begin{bmatrix} \hat{q}_1^N \\ \cdot \\ \cdot \\ \cdot \\ \hat{q}_N^N \end{bmatrix} \tag{6-56}$$

We know

$$\langle \hat{q}^j, \hat{q}^j \rangle = 1, \; j = 1, \ldots, N \tag{6-57}$$

and if we normalize equation 6-53 to

$$g = \sum_{j=1}^{N} \hat{\gamma}_j \hat{\rho}_j \hat{q}^j + \frac{Dx}{\langle x, x \rangle} \tag{6-58}$$

with

$$\hat{\rho}_j = \frac{\langle \hat{q}^j, x \rangle}{\langle x, x \rangle} \tag{6-59}$$

we restrict the range of $\hat{\rho}_j$ to ± 1.

Now $\hat{\rho}_j$ is simply the cross-correlation coefficient of vectors $\hat{q}^j$ and x. It expresses the similarity ($\hat{\rho}_j \doteq \pm 1$) or dissimilarity of $\hat{q}^j$ and x ($\hat{\rho}_j \doteq 0$). Our purpose is that if the eigenvector $\hat{q}^k$ and the data vector x are similar, then $\hat{\rho}_k$ will be large; if $\hat{q}^1, \hat{q}^2, \ldots, \hat{q}^{k-1}$ and $\hat{q}^{k+1}, \ldots, \hat{q}^N$ and x are dissimilar, all of the $\hat{\rho}_i$ will be small; then, equation 6-58 can be implemented approximately as

$$\begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_N \end{bmatrix} = \hat{\gamma}_k \hat{\rho}_k \begin{bmatrix} \hat{q}_1^k \\ \cdot \\ \cdot \\ \cdot \\ \hat{q}_N^k \end{bmatrix} + \frac{2}{\pi \langle x, x \rangle} \begin{bmatrix} x_N \\ x_{N-1} + x_{N-2} \\ \vdots \\ x_2 + x_1 \end{bmatrix} \tag{6-60}$$

In general, $g = Sx$ requires N^2 multiplications and N^2 additions to implement. However, equation 6-60 only requires N multiplications (assuming $\hat{\rho}_k$ and $\hat{\gamma}_k$ are premultiplied together and x is properly scaled) and 2N additions. 3N operations versus $2N^2$ is very significant when N is large.

We can interpret equation 6-60 as the sum of two interpolating schemes. The first part is curve fitting in a space of N dimensions; i.e., when $\hat{\rho}_k$ is large, we say the data vector x and the eigenvector q^k are very similar. Then, a part of the interpolated vector g consists of the weighted version of the "k^{th}" eigenvector, $\hat{\gamma}_k \hat{\rho}_k \hat{q}^k$. The second part of g is given by the linear interpolating scheme which consists of the scaled sum of adjacent data points, $\hat{D}x$.

Figures 6-5 and 6-6 are prints of the modified transient Whittaker matrices and the resulting eigenvalues and eigenvectors when N = 9 and 16, respectively. Figures 6-7 through 6-10 are plots of the eigenvectors when N = 16. Subroutine "INTERP" in the Appendix implements the Eigen-Filter, Cross-Correlation interpolating algorithm.

PRINT CF MODIFIED MATRIX.
MULTIPLIER= 0.637E 00 NWIND= 0

I/J	1	2	3	4	5	6	7	8	9
1	0.059	-0.067	0.077	-0.091	0.111	-0.143	0.200	-0.333	0.0
2	-0.067	0.077	-0.091	0.111	-0.143	0.200	-0.333	0.0	0.0
3	0.077	-0.091	0.111	-0.143	0.200	-0.333	0.0	0.0	-0.333
4	-0.091	0.111	-0.143	0.200	-0.333	0.0	0.0	-0.333	0.200
5	0.111	-0.143	0.200	-0.333	0.0	0.0	-0.333	0.200	-0.143
6	-0.143	0.200	-0.333	0.0	0.0	-0.333	0.200	-0.143	0.111
7	0.200	-0.333	0.0	0.0	-0.333	0.200	-0.143	0.111	-0.091
8	-0.333	0.0	0.0	-0.333	0.200	-0.143	0.111	-0.091	0.077
9	0.0	0.0	-0.333	0.200	-0.143	0.111	-0.091	0.077	-0.067

PRINT OF EIGENVALUES.
MULTIPLIER= 0.982E 00 NWIND= 0

I/J	1	2	3	4	5	6	7	8	9
1	0.956	-1.000	0.503	0.324	0.129	-0.218	-0.466	-0.411	-0.006

PRINT CF EIGENVECTCRS.
MULTIPLIER= 0.606E 00 NWIND= 0

I/J	1	2	3	4	5	6	7	8	9
1	0.190	-0.509	0.913	0.177	-1.000	0.598	0.391	0.227	0.043
2	-0.401	0.462	-0.623	0.817	-0.212	0.589	0.130	0.368	0.867
3	0.779	-0.213	0.387	0.585	0.708	0.354	-0.847	0.499	0.044
4	-0.887	-0.179	0.384	0.490	0.563	-0.235	0.522	0.726	-0.589
5	0.746	0.564	-0.382	0.234	-0.589	-0.615	0.202	0.774	-0.534
6	-0.385	-0.812	-0.214	-0.279	-0.403	-0.690	-0.654	0.644	0.559
7	-0.058	0.838	0.654	-0.891	0.237	0.070	0.056	0.646	0.563
8	0.377	-0.643	-0.726	-0.637	0.374	0.651	0.624	0.528	-0.074
9	-0.546	0.285	-0.244	-0.337	-0.397	0.745	-0.839	0.232	-0.338

6-5 Modified Transient Matrix,
Number of Windows = 0, N = 9

PRINT OF MODIFIED MATRIX.
MULTIPLIER= 0.637E 00 NWIND= 0

I/J	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	-0.032	0.034	-0.037	0.040	-0.043	0.048	-0.053	0.059	-0.067	0.077	-0.091	0.111	-0.143	0.200	-0.333	0.0
2	0.034	-0.037	0.040	-0.043	0.048	-0.053	0.059	-0.067	0.077	-0.091	0.111	-0.143	0.200	-0.333	0.0	0.0
3	-0.037	0.040	-0.043	0.048	-0.053	0.059	-0.067	0.077	-0.091	0.111	-0.143	0.200	-0.333	0.0	0.0	-0.333
4	0.040	-0.043	0.048	-0.053	0.059	-0.067	0.077	-0.091	0.111	-0.143	0.200	-0.333	0.0	0.0	-0.333	0.200
5	-0.043	0.048	-0.053	0.059	-0.067	0.077	-0.091	0.111	-0.143	0.200	-0.333	0.0	0.0	-0.333	0.200	-0.143
6	0.048	-0.053	0.059	-0.067	0.077	-0.091	0.111	-0.143	0.200	-0.333	0.0	0.0	-0.333	0.200	-0.143	0.111
7	-0.053	0.059	-0.067	0.077	-0.091	0.111	-0.143	0.200	-0.333	0.0	0.0	-0.333	0.200	-0.143	0.111	-0.091
8	0.059	-0.067	0.077	-0.091	0.111	-0.143	0.200	-0.333	0.0	0.0	-0.333	0.200	-0.143	0.111	-0.091	0.077
9	-0.067	0.077	-0.091	0.111	-0.143	0.200	-0.333	0.0	0.0	-0.333	0.200	-0.143	0.111	-0.091	0.077	-0.067
10	0.077	-0.091	0.111	-0.143	0.200	-0.333	0.0	0.0	-0.333	0.200	-0.143	0.111	-0.091	0.077	-0.067	0.059
11	-0.091	0.111	-0.143	0.200	-0.333	0.0	0.0	-0.333	0.200	-0.143	0.111	-0.091	0.077	-0.067	0.059	-0.053
12	0.111	-0.143	0.200	-0.333	0.0	0.0	-0.333	0.200	-0.143	0.111	-0.091	0.077	-0.067	0.059	-0.053	0.048
13	-0.143	0.200	-0.333	0.0	0.0	-0.333	0.200	-0.143	0.111	-0.091	0.077	-0.067	0.059	-0.053	0.048	-0.043
14	0.200	-0.333	0.0	0.0	-0.333	0.200	-0.143	0.111	-0.091	0.077	-0.067	0.059	-0.053	0.048	-0.043	0.040
15	-0.333	0.0	0.0	-0.333	0.200	-0.143	0.111	-0.091	0.077	-0.067	0.059	-0.053	0.048	-0.043	0.040	-0.037
16	0.0	0.0	-0.333	0.200	-0.143	0.111	-0.091	0.077	-0.067	0.059	-0.053	0.048	-0.043	0.040	-0.037	0.034

PRINT OF EIGENVALUES.
MULTIPLIER= 0.119E 01 NWIND= 0

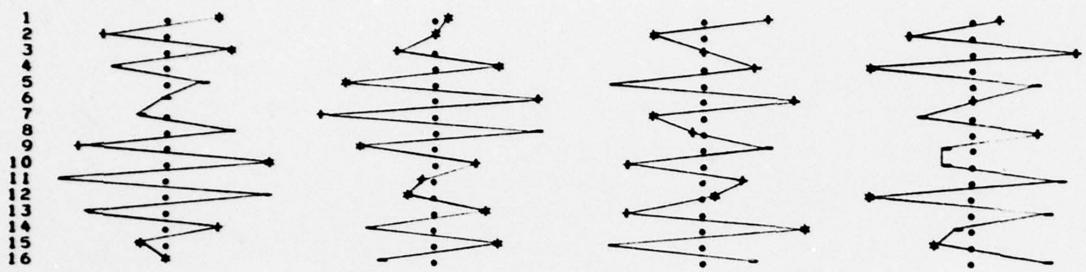
I/J	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1.000	-0.985	0.731	-0.706	0.465	-0.467	0.333	0.249	0.217	0.112	-0.354	-0.298	-0.259	-0.066	-0.170	0.005

PRINT OF EIGENVECTORS.
MULTIPLIER= 0.463E 00 NWIND= 0

I/J	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.427	0.088	0.566	0.260	0.489	0.820	-0.222	0.385	-0.374	0.718	0.103	0.581	0.993	0.982	-0.136	0.068
2	-0.498	0.006	-0.429	-0.497	-0.010	-0.740	-0.404	0.523	-0.587	0.527	0.246	0.510	-0.470	0.261	-1.000	0.876
3	0.533	-0.269	0.029	0.871	-0.688	0.652	-0.645	0.798	0.548	-0.029	0.284	0.462	-0.524	-0.723	-0.452	0.036
4	-0.448	0.524	0.462	-0.838	0.754	0.144	-0.736	0.275	-0.037	-0.597	0.490	0.749	0.018	-0.679	0.251	-0.621
5	0.274	-0.740	-0.781	0.479	-0.150	-0.490	-0.723	-0.378	-0.636	-0.439	0.605	0.489	-0.009	0.602	0.665	-0.516
6	-0.034	0.873	0.770	0.049	-0.524	0.226	-0.606	-0.705	0.090	-0.024	0.607	-0.310	-0.824	0.708	0.375	0.566
7	-0.235	-0.899	-0.432	-0.463	0.562	0.454	-0.485	-0.491	0.968	0.592	0.691	-0.394	0.246	-0.189	0.190	0.592
8	0.495	0.810	-0.065	0.548	0.069	-0.793	-0.186	-0.453	-0.153	0.541	0.800	-0.381	0.811	-0.811	-0.389	-0.127
9	-0.708	-0.620	0.475	-0.261	-0.715	0.461	0.078	0.030	-0.627	-0.029	0.717	-0.782	0.209	0.049	-0.736	-0.805
10	0.842	0.357	-0.593	-0.240	0.705	0.220	0.327	0.619	0.147	-0.793	0.694	-0.747	-0.330	0.578	-0.495	-0.074
11	-0.881	-0.064	0.359	0.685	-0.050	-0.527	0.523	0.863	0.424	-0.526	0.727	0.014	0.539	0.198	0.559	0.600
12	0.820	-0.213	0.114	-0.834	-0.562	0.176	0.731	0.314	-0.683	0.297	0.619	0.215	-0.048	-0.546	0.845	0.509
13	-0.671	0.429	-0.597	0.611	0.498	0.537	0.720	0.005	-0.247	0.896	0.461	0.224	-0.839	-0.098	0.341	-0.538
14	0.459	-0.548	0.854	-0.126	0.200	-0.754	0.685	-0.368	0.748	0.261	0.446	0.699	-0.498	0.334	-0.455	-0.561
15	-0.228	0.540	-0.773	-0.310	-0.784	0.515	0.541	-0.650	0.486	-0.477	0.307	0.915	0.597	0.201	-0.481	0.081
16	-0.011	-0.445	0.388	0.607	0.732	0.425	0.363	-0.812	-0.766	-0.764	0.102	0.257	-0.172	-0.505	-0.489	0.818

6-6 Modified Transient Matrix,
Number of Windows = 0, N = 16

PLOT OF EIGENVECTORS.
MULTIPLIER= 0.463E 00 NWIND= 0
COLUMNS PLOTTED. 1. 2. 3. 4.

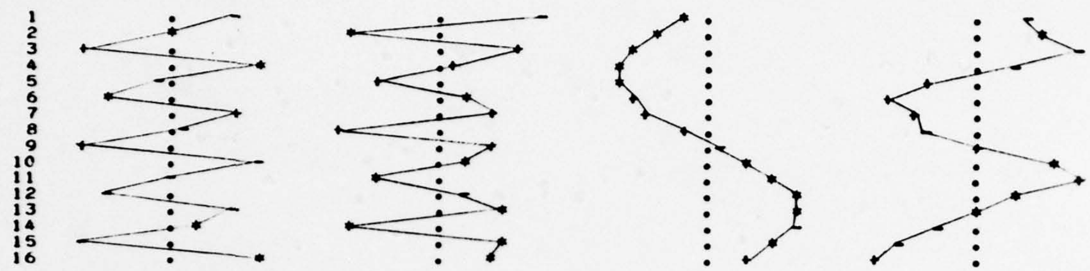


LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 19

6-7 Eigenvectors (1-4) of Modified Matrix, N = 16

PLOT OF EIGENVECTORS.
MULTIPLIER= C.463E 00 NWIND= 0
COLUMNS PLOTTED. 5. 6. 7. 8.

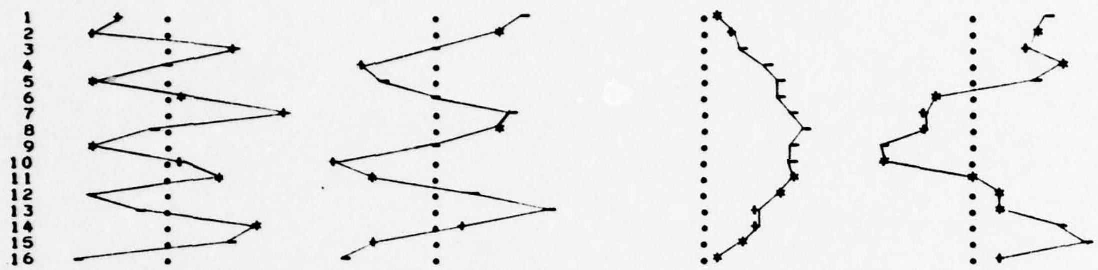


LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-FEAK PRINT CELLS= 19

6-8 Eigenvectors (5-8) of Modified Matrix, N = 16

PLOT OF EIGENVECTORS.
MULTIPLIER= 0.463E 00 NWIND= 0
COLUMNS PLOTTED. 9. 10. 11. 12.

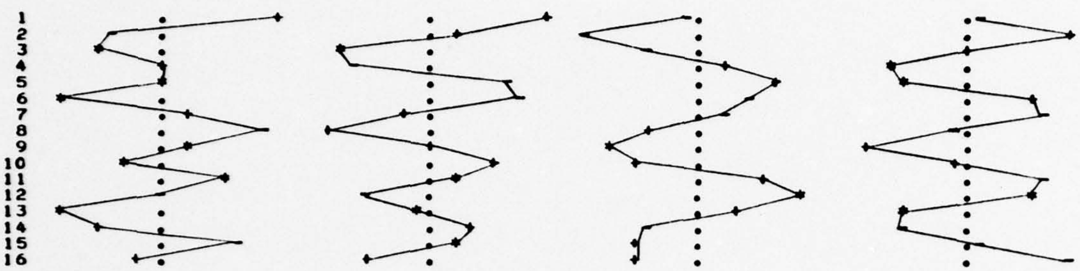


LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 19

6-9 Eigenvectors (9-12) of Modified Matrix, N = 16

PLOT OF EIGENVECTORS.
MULTIPLIER= 0.463E 00 NWIND= 0
COLUMNS PLOTTED. 13. 14. 15. 16.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 19

6-10 Eigenvectors (13-16) of Modified Matrix, N = 16

CHAPTER VII

ERRORS

The relative ease in deriving error bounds for matrix equations was one of our motivating factors in developing the Whittaker matrix process. We begin this chapter by briefly reviewing vector and matrix norms and condition numbers, and then use these concepts in deriving various error bounds in Sections 2 through 5. In Section 2 we derive two expressions which describe the effects of the truncated Whittaker matrix on the interpolated data. First, we bound the error caused by the truncation and then present a formula which predicts the sensitivity of the interpolants to data outside the interpolating interval. In Section 3 we develop an expression for the noise-to-signal ratio of the interpolated data. We show simply that this ratio is the magnified noise-to-signal ratio of the original data. The magnification factor is shown to be the condition number of the Whittaker matrix. We discuss the effects of the computer hardware on the interpolants in Section 4. By assuming the worst possible computer roundoff errors, we show that interpolation with the Whittaker matrices is relatively immune to roundoff problems. In the final section, we derive error expressions for the Eigen-Filter, Cross-Correlation Algorithm described in Chapter VI.

7.1 Norms and Condition Numbers

The norm of a vector, signified by $||\cdot||$, is a functional from vectors to non-negative reals which satisfies the following properties [33, p. 163]

$$||x|| > 0 \qquad (7\text{-}1)$$

$$||x|| = 0,\ x = 0 \qquad (7\text{-}2)$$

$$||\lambda x|| = |\lambda|\ ||x|| \qquad (7\text{-}3)$$

$$||x + y|| \leq ||x|| + ||y|| \qquad (7\text{-}4)$$

As used in this chapter, we define the Holder, or "p", norms as [33, p. 166]

$$||x||_p = \left(\sum_{i=1}^{N} |x_i|^p\right)^{1/p}, \quad 1 \leq p \leq \infty \qquad (7\text{-}5)$$

Then,

$$||x||_1 = \sum_{i=1}^{N} |x_i| \qquad (7\text{-}6)$$

$$||x||_2 = \left(\sum_{i=1}^{N} |x_i|^2\right)^{1/2} \qquad (7\text{-}7)$$

$$||x||_\infty = \underset{i}{\text{MAX}} |x_i| \qquad (7\text{-}8)$$

Given any vector norm, the subordinate matrix norm is defined as [39, p. 56]

$$||S|| = \max_{||x||=1} \frac{||Sx||}{||x||} \qquad (7\text{-}9)$$

In addition to the properties in equations 7-1 through 7-4 (replace x with S), matrix norms satisfy the following

$$||Sx|| \leq ||S||\ ||x|| \qquad (7\text{-}10)$$

which follows from equation 7-9.

Some specific norms have additional useful properties. If S is symmetric and Q is orthogonal (i.e., $Q^TQ = QQ^T = I$) we have [33, p. 308]

$$||S||_2 = ||Q\Gamma Q^T||_2 = ||\Gamma||_2 = |\gamma_{max}| \qquad (7\text{-}11)$$

where the γ_i are eigenvalues of S, and Γ is the diagonal matrix of all eigenvalues. We also have the "Frobenius" norm [33, p. 173] which can be defined for a real symmetric matrix as

$$||S||_F = \left(\sum_i \sum_j s_{ij}^2\right)^{1/2} = \left(\sum_k t_{kk}\right)^{1/2} \qquad (7\text{-}12)$$

where

$$T = S^TS = (Q\Gamma Q^T)^T(Q\Gamma Q^T) = Q\Gamma^2Q^T \qquad (7\text{-}13)$$

Then

$$\sum_k t_{kk} = \sum_i \gamma_i^2 \qquad (7\text{-}14)$$

and we have the useful results

$$||S||_2 = |\gamma_{max}| \leq ||S||_F = (\sum_i \gamma_i^2)^{1/2} \leq \sqrt{N}\,|\gamma_{max}| \tag{7-15}$$

The condition number of a matrix is defined [33, p. 190] as the product of the norms of the matrix and its inverse

$$k_p(s) = ||S||_p \; ||S^{-1}||_p \tag{7-16}$$

When S is symmetric and p = 2, we have using equation 7-11

$$k_2(s) = |\frac{\gamma_{max}}{\gamma_{min}}| \tag{7-17}$$

We note that large $k_2(s)$ implies $\gamma_{min} \to 0$. In other words, "poor" or a large condition is predicated on "closeness" to singularity. Also, when $S^2 = I$, we have $k_2(s) = 1$. Thus, involutory or orthogonal matrices are "perfectly" conditioned with respect to the 2 norm.

In the remainder of this chapter we drop the subscript p from our derivations. Unless otherwise indicated, we will always restrict our discussion to forms involving the "2" norm.

7.2 Truncation Errors

Truncation errors result when we truncate the infinite Whittaker summation and formulate the transient Whittaker matrix interpolating equation

$$g = Sx \tag{7-18}$$

Then

$$x = S^{-1}g \tag{7-19}$$

and

$$||x|| \leq ||S^{-1}|| \ ||g|| \tag{7-20}$$

From equation 4-13 write

$$f_i^N = \frac{2}{\pi} \sum_{p=-i}^{N-i-1} \frac{(-1)^p}{2p+1} x_{i+p+1} \tag{7-21}$$

where the superscript "N" indicates that we have truncated the Whittaker summation to N terms. Now, suppose we interpolate with an N + L point scheme and formulate the difference

$$|f_i^{N+L} - f_i^N| = \frac{2}{\pi}\left| \sum_{p=N-i}^{N-i-1+L} \frac{(-1)^p}{2p+1} x_{i+p+1} \right| \tag{7-22}$$

Define

$$\Delta f_i = \lim_{L\to\infty} |f_i^{N+L} - f_i^N| \tag{7-23}$$

Then, using the Cauchy Schwartz inequality

$$\Delta f_i \leq \frac{2}{\pi} \sum_{p=N-i}^{\infty} \left|\frac{1}{2p+1} x_{i+p+1}\right|$$

$$\leq \frac{2}{\pi}\left[\sum_{p=N-i}^{\infty}\left(\frac{1}{2p+1}\right)^2\right]^{1/2}\left[\sum_{p=N+1}^{\infty}x_p^{\ 2}\right]^{1/2} \qquad (7\text{-}24)$$

From section 3.1 we assume that a data set $\{x_i, i = 1, 2, ..., \infty\}$ must be square summable in order to use Whittaker interpolation. Thus, we assume the norm of the data outside our interpolating interval is equal to some constant α times the norm of the data vector x; i.e.,

$$\left[\sum_{p=N+1}^{\infty}x_p^{\ 2}\right]^{1/2} \leq \alpha\left[\sum_{i=1}^{N}x_i^2\right]^{1/2} = \alpha||x|| \qquad (7\text{-}25)$$

Using this inequality and equation 7-20, we have

$$\Delta f_i \leq \frac{2}{\pi}\alpha\left[\sum_{p=N-i}^{\infty}\left(\frac{1}{2p+1}\right)^2\right]^{1/2}||S^{-1}||\ ||g|| \qquad (7\text{-}26)$$

Recognizing that $\Delta f_i = \Delta g_{N-i+1}$, we can state a bound for the normalized truncation error as

$$e_{N-i+1} = \frac{\Delta g_{N-i+1}}{||g||} \leq \frac{2}{\pi}\alpha\left[\sum_{p=N-i}^{\infty}\left(\frac{1}{2p+1}\right)^2\right]^{1/2}||S^{-1}|| \qquad (7\text{-}27)$$

or

$$e_i \leq \frac{2}{\pi}\alpha\left[\left(\sum_{p=0}^{\infty} - \sum_{p=0}^{i-2}\right)\left(\frac{1}{2p+1}\right)^2\right]^{1/2}\left|\frac{1}{\gamma_{min}}\right| \qquad (7\text{-}28)$$

$$e_i \leq \frac{2}{\pi}\alpha[1.2337 \ldots - \sum_{p=0}^{i-2}(\frac{1}{2p+1})^2]^{1/2} |\frac{1}{\gamma_{min}}| \quad (7\text{-}29)$$

Easily computed for N = 16, we have

$$e_1 \leq \frac{2}{\pi}\alpha[1.2337 - 0.0]^{1/2} \quad \frac{1}{.385} = 1.837\alpha \quad (7\text{-}30)$$

and

$$e_{16} \leq \frac{2}{\pi}\alpha[1.2337 - 1.2170]^{1/2}\frac{1}{.385} = .213\alpha \quad (7\text{-}31)$$

If, for instance, we assume α is .1, $e_i \leq .184$.

Equation 7-27 can be put in a more convenient form if we approximate the summation with the integral

$$\sum_{p=2N-i}^{\infty}(\frac{1}{2p+1})^2 \doteq \int_{N-i}^{\infty} \frac{1}{(2x+1)^2}dx = \frac{1}{4N - 4i + 2}$$

(7-32)

Then, the error expression becomes

$$e_i \leq \frac{2}{\pi}\alpha \; [\frac{1}{4i - 2}]^{1/2} \; |\frac{1}{\gamma_{min}}| \quad (7\text{-}33)$$

For N = 16, $e_1 \leq .210\alpha$ which agrees with equation 7-31.

Another factor relating to truncation error is a sensitivity factor. First, we write equation 7-18 with superscripts indicating the number of data points used in the interpolation. Thus

$$g^N = S^N x^N \quad (7\text{-}34)$$

If we add one more data point to the interpolation scheme, we can write

$$g^{N+1} = \begin{bmatrix} \hat{g}^{N+1} \\ g_{N+1} \end{bmatrix} \begin{bmatrix} S^N & \hat{s}^{N+1} \\ \hat{s}^{(N+1)T} & s^{N+1}_{N+1} \end{bmatrix} \begin{bmatrix} x^N \\ x_{N+1} \end{bmatrix} \qquad (7\text{-}35)$$

Subtracting g^N from $\hat{g}^{N+1}$ and multiplying the absolute difference by $||x^N||$ we can write

$$|\hat{g}^{N+1} - g^N| \; ||x^N|| \leq |\hat{s}^{N+1} x_{N+1}| \; ||S^{-1}|| \; ||g^N|| \qquad (7\text{-}36)$$

For the i^{th} interpolant, define

$$\delta_i = \frac{|g_i^{N+1} - g_i^N|}{||g^N||} \qquad (7\text{-}37)$$

then

$$\delta_i \leq |s_i^{N+1}| \frac{|x_{N+1}|}{||x^N||} \frac{1}{|\gamma_{min}|} \qquad (7\text{-}38)$$

If we use equation 4-29 for the transient case

$$\delta_i \leq \frac{2}{\pi} \left|\frac{1}{3 - 2i}\right| \frac{|x_{N+1}|}{||x^N||} \frac{1}{|\gamma_{min}|} \qquad (7\text{-}39)$$

Then, for $i = N$

$$\delta_N \leq \frac{2}{\pi} \left|\frac{1}{3 - 2N}\right| \frac{|x_{N+1}|}{||x^N||} \frac{1}{|\gamma_{min}|} \qquad (7\text{-}40)$$

and for $N = 16$, we can now state the normalized sensitivity of the first of 16 interpolants to the change caused by adding one more data point

$$\delta_{16} \leq \frac{2}{\pi} \frac{1}{29} \frac{|x_{N+1}|}{||x^N||} \frac{1}{.385} \qquad (7\text{-}41)$$

or

$$\delta_{16} \leq .057 \frac{|x_{N+1}|}{||x^N||} \qquad (7\text{-}42)$$

Loosely stated, if a 17^{th} data point is about the size of the norm of the first 16 points used in the interpolation (a rather horrible situation) then we should expect up to a 6 percent change in the interpolant if we go to a 17 point scheme. However, if we assume x_{17} is only about .1 times the norm of the first 16 points (still a bad situation) then the interpolant is relatively "insensitive" to the 17^{th} point; i.e., $\delta_{16} \leq .006$.

7.3 Errors Due to Noisy Data

Suppose the data vector in equation 7-20 is corrupted with noise. We say

$$(g + \Delta g) = S(x + \Delta x) \qquad (7\text{-}43)$$

where Δx is the vector of instantaneous noise values and Δg is the resultant change in the vector of interpolants. Now we can write

$$\Delta g = S \Delta x \qquad (7\text{-}44)$$

and

$$||\Delta g|| \leq ||S|| \ ||\Delta x|| = |\gamma_{max}| \ ||\Delta x|| \tag{7-45}$$

since

$$||x|| \leq ||S^{-1}|| \ ||g|| = \frac{1}{|\gamma_{min}|}||g|| \tag{7-46}$$

we have

$$||\Delta g|| \ ||x|| \leq ||S|| \ ||S^{-1}|| \ ||\Delta x|| \ ||g||$$

$$= k(s)||\Delta x|| \ ||g|| \tag{7-47}$$

or finally,

$$\frac{||\Delta g||}{||g||} \leq k(s) \ \frac{||\Delta x||}{||x||} = \left|\frac{\gamma_{max}}{\gamma_{min}}\right| \frac{||\Delta x||}{||x||} \tag{7-48}$$

where k(s) is given by equation 7-16.

The ratio $||\Delta x||/||x||$ is the ratio of the square root of noise power to square root of signal power, i.e., roughly the reciprocal signal-to-noise ratio, $(S/N)^{-1}$. Thus, equation 7-48 implies that Whittaker interpolation can magnify the N/S ratio by k(s). For the periodic symmetric nonsingular Whittaker matrix with N = 9, k(s) = 1, thereby preserving N/S. For the 16 point transient interpolation, k(s) = 2.6, which means that the interpolants have gained less than 8.3db in noise. The value for k(s) for any specific transient problem can be computed by using subroutine SYMEIG in the Appendix.

7.4 Machine Roundoff Errors

We know, in general, that the elements of S and x cannot be exactly represented internally in a computer. We can express the resulting errors in g as follows

$$(g + \Delta g) = (S + \Delta S)(x + \Delta x) \qquad (7\text{-}49)$$

where ΔS and Δx are now the machine errors caused by roundoff. Then

$$(g + \Delta g) = Sx + [S\Delta x + \Delta S(x + \Delta x)] \qquad (7\text{-}50)$$

or,

$$\Delta g = S\Delta x + \Delta S(x + \Delta x) \qquad (7\text{-}51)$$

Proceeding as before, since $||x|| \leq ||S^{-1}||\ ||g||$, we write

$$||\Delta g||\ ||x|| \leq ||S\Delta x + \Delta S(x + \Delta x)||\ ||S^{-1}||\ ||g|| \qquad (7\text{-}52)$$

and using equations 7-4 and 7-10,

$$\frac{||\Delta g||}{||g||} \leq k(s)\left[\frac{||\Delta x||}{||x||} + \frac{||\Delta S||}{||S||}\frac{||x||}{||x||} + \frac{||\Delta S||}{||S||}\frac{||\Delta x||}{||x||}\right] \qquad (7\text{-}53)$$

On the IBM 360/67 using single precision arithmetic, roundoff can be expressed as

$$\hat{g}_i = g_i(1 + \delta_i), \ |\delta_i| \le 16^{-5} \tag{7-54}$$

If we assume the worst possible case for $||\Delta x||$ and $||\Delta S||$ we can write

$$||\Delta x|| = ||16^{-5}x|| = 16^{-5}\ ||x|| \tag{7-55}$$

Also,

$$||\Delta S|| \le ||\Delta S||_F = 16^{-5}||S||_F \tag{7-56}$$

Using equation 7-15,

$$||\Delta S||_F \le 16^{-5}(\sum \gamma_i^2)^{1/2} \le 16^{-5}\sqrt{N}|\gamma_{max}| \tag{7-57}$$

But $|\gamma_{max}|$ is simply $||S||$; therefore,

$$||\Delta S|| \le 16^{-5}\sqrt{N}||S|| \tag{7-58}$$

Substituting in equation 7-53

$$\frac{||\Delta g||}{||g||} \le k(s)[16^{-5} + \sqrt{N}\ 16^{-5} + \sqrt{N}\ 16^{-10}] \tag{7-59}$$

For moderate N, then, we can reasonably expect

$$\frac{||\Delta g||}{||g||} \dot{\le} k(s)\sqrt{N}\ 16^{-5} = |\frac{\gamma_{max}}{\gamma_{min}}|\sqrt{N}\ 16^{-5} \tag{7-60}$$

For the transient Whittaker matrix and N = 16, equation 7-60 implies that the norm of the normalized error in the interpolants due to roundoff is less than 10^{-5}.

7.5 Approximation Errors

In the Eigen-Filter, Cross-Correlation Algorithm, we produce an approximation g* to g. If we solve the following equation for x*,

$$x^* = S^{-1}g^* \tag{7-61}$$

we find that x - x* is in general non-zero. Define a vector of residuals

$$r = x - x^* \tag{7-62}$$

Then,

$$r = x - S^{-1}g^* \tag{7-63}$$

and multiplying both sides by S, we have

$$Sr = Sx - g^* = g - g^* \tag{7-64}$$

But g - g* is the error due to the approximation. Then, define an error vector

$$e = Sr \tag{7-65}$$

Proceeding as in previous sections

$$||e|| \leq ||S|| \; ||r|| \tag{7-66}$$

Using $||x^*|| \leq ||S^{-1}|| \; ||g^*||$, we express

$$||e|| \; ||x^*|| \leq ||S|| \; ||S^{-1}|| \; ||r|| \; ||g^*|| \tag{7-67}$$

or

$$\frac{||e||}{||g^*||} \leq k(s) \frac{||r||}{||x^*||} = \left|\frac{\gamma_{max}}{\gamma_{min}}\right| \frac{||r||}{||x^*||} \tag{7-68}$$

This "nice" expression allows the norm of the normalized error in g* to be bounded without really knowing g; i.e., compute g* as described in Chapter VI, use equations 7-61 and 7-62 to find x* and r; and, finally, use equation 7-68 to bound the errors in g*. For the transient Whittaker matrix and N = 16, the norm of normalized errors due to approximation is less than 2.6 times the norm of normalized residuals in x.

CHAPTER VIII

RESULTS

The Appendix is the listing of a lengthy computer program which implements the major algorithms of this dissertation. The program evolved over a year's work and is somewhat elaborate as to documentation. We believe that reading Section 1 of this chapter, wherein we briefly describe the program, and then reading the listing comments on input and output parameters, will enable one to run the program if so desired. Section 2 of this chapter is basically a compendium of results from the program and a discussion of their significance. Ten rather difficult to interpolate functions were used to produce the various figures in this section. The final Section 3 is a summary of what we set out to do in the dissertation and our own assessment of what we actually accomplished.

8.1 Description of Computer Program

The techniques discussed throughout the dissertation were programmed as docummented in the Appendix. Basically, we programmed a three step effort: on the first pass, the transient symmetric Whittaker matrix is formed in subroutine MATRIX, and subroutine SYMEIG is used to find the eigenvalues and eigenvectors. Subroutine SIGNAL produces N data points from some band-limited function and subroutine INTERP produces a vector of interpolants by implementing the orthogonal similarity transformation of equation 6-58. The routine also produces a vector of correlation coefficients per equation 6-59. Various prints and plots are produced by subroutines PRINT and PLOT to document the first pass.

Before the second step, one analyzes the results of the first pass and selects eigenvectors that "strongly" correlated with the original data vector. We considered a correlation coefficient of .2 or greater as being significant. These eigenvector numbers are input for the second step. This run basically repeats the first step except that only the selected eigenvectors are used in INTERP to produce the vector of interpolants. Again, various prints and plots are produced to document the second pass.

The third step is for the periodic symmetric Whittaker matrix. Subroutine MATRIX produces this cyclical form of the Whittaker matrix and the other subroutines follow as in the transient case: subroutine INTERP interpolates using the orthogonal similarity transformation, correlation coefficients are produced, and prints and plots are produced. In addition, subroutine FFTINT is invoked to implement the equivalence transformation discussed in Chapter V. Both the INTERP and FFTINT routines are used to document that they produce the same results (except for roundoff) although being completely different algorithms.

In the final runs for this dissertation, all three steps were run together since prior experimentation determined which eigenvectors were to be used for the second step. The compile and run times are documented in Figure 8-11. In addition, the program prints a total of 2200 lines when the program listing is requested, or about 1300 lines for all three steps when only results are required. Total cards in the program Fortran deck are about 850, including comments.

8.2 Comparison of Relative Errors

Figures 8-1 to 8-10 depict the results of running the program in the Appendix on 10 selected signals. (The signals are shown as solid lines.) The "a" part of the Figures are plots of the interpolants using the transient technique of Chapter VI, and also show the correlation coefficients from equation 6-59. The "b" parts show plots of the interpolants using selected eigenvectors and plots of the interpolants from the FFT algorithm of Chapter V. The final "c" parts of the Figures are tabular summaries of the data used to generate the plots. The upper left table is for the full transient interpolation scheme and the next table is for the selected eigenvector approach. The lower two tables are for the symmetric periodic matrix and FFT algorithms.

The lower two tables are both included to show that there is no discernible difference in the matrix and FFT approaches. This is because the orthogonal similarity transformation on the real symmetric periodic Whittaker matrix, and the complex unitary equivalence transformation implemented via the FFT must multiply to the same matrix except for roundoff error. When the FFT and periodic matrix algorithms are used on a perfectly band-limited function, Figure 8-11 shows that the norm of roundoff error, $||e||$, from the matrix technique is an order of magnitude greater than for the FFT algorithm.

Figures 8-11 through 8-13 are tabular summaries of data from Figures 8-1 through 8-10. Figure 8-11 is a summary of the norms of the original data, the various interpolants and the errors, plus a listing of the program compilation times and run times. All the data, for instance in Figures 8-1a, b, and c, were produced during one computer run at a cost of 16.8 seconds compile time and 11.19 seconds run time on the UNM IBM 360-67 using the standard Fortran IV-G compiler. Figure 8-12 has scale factors applied (listed in heading information in Figures 8-1c to 8-10c) to the maximum and minimum errors so absolute error comparisons can be made. Finally, Figure 8-13 is a comparison of the relative maximum errors in the interpolants and is the most important of the three summaries.

Figure 8-13 also lists a percentage factor for how much transient signal lies outside the region of interpolation;

that is, given that each data vector x behaves inside the region of interpolation as depicted by the solid curve in Figures 8-1a to 8-10a, how much more signal must there be in order that x come from a band-limited process? This amount is expressed in Figure 8-13 as a percentage of $||x||$ and is the "α" factor derived in equation 7-31 times 100.

Figure 8-13 summarizes several interesting results. First, and obvious, small relative error implies small α. In other words, if we properly sample a reasonably band-limited phenomena and use 2TW samples in the transient interpolation scheme, then we should expect small errors in the interpolants and little significant data outside the interpolation interval. This result is clearly demonstrated in Signals 1, 2, 3, 4, 5, 8, and 9 where relative errors are less than .091 and energy outside the interpolation interval (remember spread over infinity) is less than 43% of the energy inside the interval. Second, FFT (or the equivalent symmetric periodic matrix) interpolation reduces the relative error when: (a) the periodic extension of the transient phenomena forms a perfectly band-limited process; e.g., the extension of the single pulse of a sinusoidal wave form in Signal 1; (b) the periodic extension of the transient phenomena reduces the severity of a discontinuity; i.e., the extension of the half cycle sine wave plus d.c. offset in Signal 3. The Figure also shows that FFT interpolation can increase the error when the phenomena begins and ends at significantly different levels as in Signal 4. This obtains from forcing the periodic extension and thereby forcing the last interpolating point to be midway between the discontinuity. Finally, and surprisingly so, the interpolants computed by the selected eigenvector algorithm are generally better in the sense of smaller relative errors than those using the full transient interpolation. This can be seen for Signals 3, 4, 6, 8, and 9. A visual improvement is also noticed for Signal 7 (Figure 8-7b) where the oscillatory overshoot of full transient interpolation is significantly reduced by the eigenfilter techniques. The maximum relative error, though, for this example is a little larger than for full transient interpolation. Actually, to this author, every example used visually appears - overall - as good as or better when using the selected eigenvector approach. This assertion is verified by Figure 8-11 wherein the norms of the total errors using selected eigenvectors are less than for the transient errors for all signals except 1, 2, and 5 and approximately equal in these three examples.

8.3 Conclusions

Our main intent for this dissertation was to describe interpolation as a matrix process and to present algorithms which implement the matrix techniques as an alternative to simple linear interpolation algorithms. Our purpose was to gain additional insight into what interpolation really does to the engineering band-limited function.

In Chapters IV, V, and VI we did describe interpolation in terms of the Whittaker matrix processes. We were able to derive closed form expressions for elements of the periodic symmetric matrix in terms of simple cosecant and cotangent functions. In Chapter V we were able to show the complete agreement of a symmetric matrix decomposition description of interpolation and the Fast Fourier Transform (FFT) implementation so in vogue with the engineers. In Chapter VI we also described a new matrix algorithm for curve fitting via the Eigen-Filter, Cross-Correlation Algorithm. Here, we showed that the correlation coefficient is the indicator of goodness of fit of a curve to data, much akin to the way we say a minimum sum of squared residuals is a measure of goodness of fit of a curve in the least squares sense.

The new insights into interpolation are implicit in the linear algebra interpretations of matrix transformations on linear vector spaces. Given 2TW samples and the assurances of the data gatherer that the samples are not aliased, we can now view interpolation as a mapping of the data vector of 2TW components into an interpolant vector of 2TW components. Of particular importance is that by modifying the transient Whittaker matrix as described in Chapter VI, we can generate an orthogonal similarity transformation which has distinct eigenvalues and a complete set of unique eigenvectors. Any function in the space of 2TW dimension is then representable as a linear combination of the 2TW basis vectors - the eigenvectors. We used 10 rather "nasty" functions, in the Fourier sense, to demonstrate the viability of using selected eigenvectors for interpolation. We also showed in Chapter VII that error bounds for matrix equations can be easily and clearly established in terms of matrix and vector condition numbers and norms.

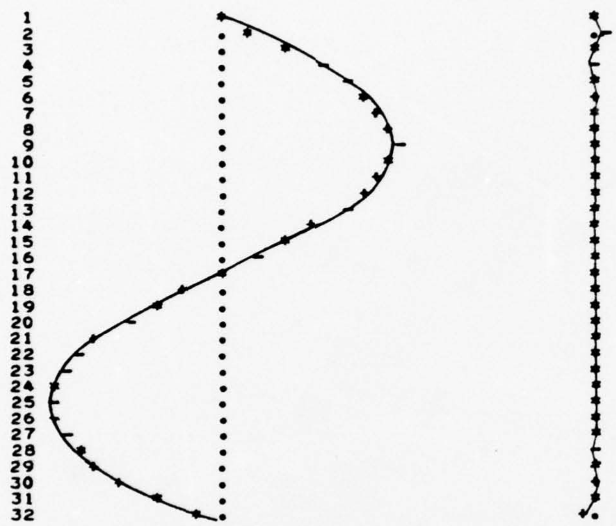
As an alternative to linear approximation approaches to interpolation, we presented a whole appendix of computer programs to implement several exact interpolation schemes. We do not claim our code is optimum, but merely that it does work as shown by the numerous examples in section 2 of this chapter.

In conclusion, we point out that any finite impulse response, nonrecursive digital filter can be formulated either as a transient or periodic symmetric matrix process as we have done. Possibilities for future work abound: by replacing our subroutine MATRIX with one that generates the appropriate matrix (or its inverse) for the problem at hand, we can describe filtering or <u>deconvolution</u> (matrix inverse) in terms of similarity transformations and matrix norms.

PLOT OF MATRIX INTERPOLATION
MULTIPLIER= 0.100E 01 NWIND= 0
EIGENVECTORS USED IN THE RECONSTRUCTION
1 2 3 4 5 6 7 8

9 10 11 12 13 14 15 16

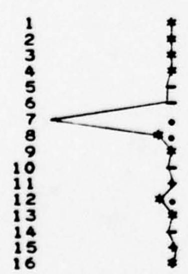
COLUMN 1 IS INTERPOLATED DATA
COLUMN 2 IS THE ERROR
COLUMNS PLOTTED. 1. 2.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27

PLOT OF CORRELATION COEFFICIENTS
MULTIPLIER= 0.100E 01 NWIND= 0
COLUMNS PLOTTED. 1.

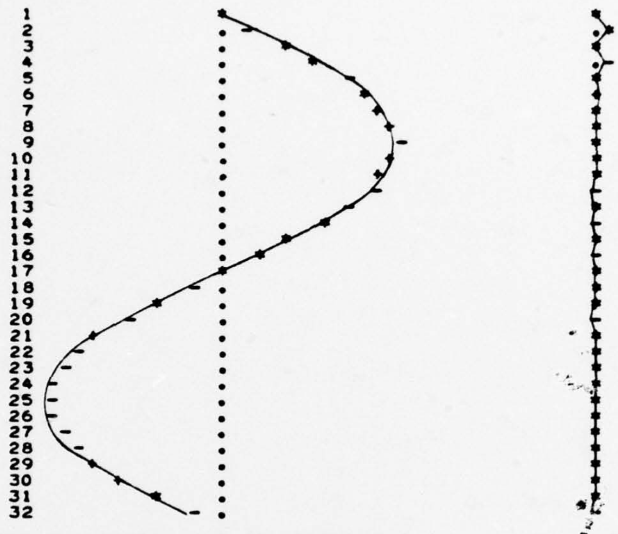


LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

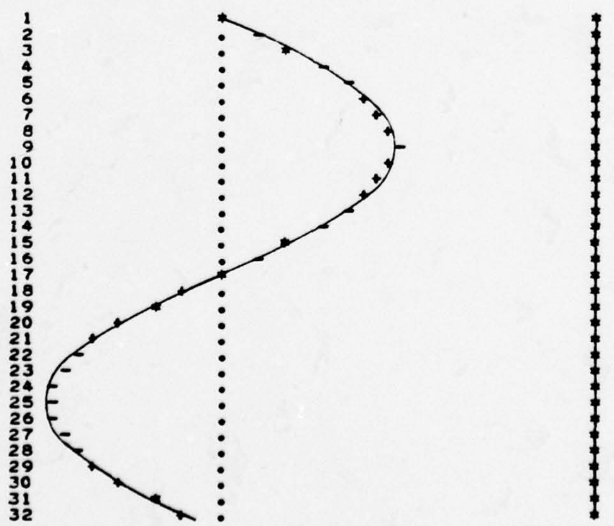
NUMBER OF PEAK-TO-PEAK PRINT CELLS= 19

8-1a Transient Interpolation and Correlation Coefficients - Signal 1

```
PLOT OF MATRIX INTERPOLATION
MULTIPLIER= 0.100E 01 NWIND=  0
EIGENVECTORS USED IN THE RECONSTRUCTION
  0  0  0  0  0  0  7  0

  0  0  0  0  0  0  0  0

COLUMN 1 IS INTERPCLATED DATA
COLUMN 2 IS THE ERROR
COLUMNS PLOTTED.  1.  2.
```



```
LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27

PLOT OF FFT INTERPOLATION
MULTIPLIER= 0.100E 01 NWIND=999
COLUMN 1 IS INTERPCLATED DATA
COLUMN 2 IS THE ERROR
COLUMNS PLOTTED.  1.  2.
```



```
LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27
```

8-1b Eigenfilter and FFT Interpolation - Signal 1

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.100E 01 NWIND= 0
NORM(X)= 0.283E 01 NORM(G)= 0.283E 01 NORM(E)= 0.708E-01

I/J	1	2
1	0.0	0.0
2	0.157	0.038
3	0.383	0.0
4	0.576	-0.020
5	0.707	0.0
6	0.817	0.014
7	0.924	0.0
8	0.992	-0.011
9	1.000	0.0
10	0.971	0.010
11	0.924	0.0
12	0.840	-0.009
13	0.707	0.0
14	0.547	0.008
15	0.383	0.0
16	0.203	-0.008
17	0.000	0.0
18	-0.203	0.008
19	-0.383	0.0
20	-0.547	-0.008
21	-0.707	0.0
22	-0.840	0.009
23	-0.924	0.0
24	-0.971	-0.010
25	-1.000	0.0
26	-0.992	0.011
27	-0.924	0.0
28	-0.817	-0.014
29	-0.707	0.0
30	-0.576	0.020
31	-0.383	0.0
32	-0.157	-0.038

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.100E 01 NWIND= 0
NORM(X)= 0.283E 01 NORM(G)= 0.284E 01 NORM(E)= 0.104E 00

I/J	1	2
1	0.0	0.0
2	0.126	0.070
3	0.383	0.0
4	0.518	0.038
5	0.707	0.0
6	0.816	0.016
7	0.924	0.0
8	0.991	-0.010
9	1.000	0.0
10	0.987	-0.006
11	0.924	0.0
12	0.868	-0.037
13	0.707	0.0
14	0.588	-0.032
15	0.383	0.0
16	0.218	-0.023
17	0.000	0.0
18	-0.183	-0.012
19	-0.383	0.0
20	-0.536	-0.019
21	-0.707	0.0
22	-0.841	0.010
23	-0.924	0.0
24	-0.985	0.309
25	-1.000	0.0
26	-0.985	0.004
27	-0.924	0.0
28	-0.829	-0.003
29	-0.707	0.0
30	-0.562	0.007
31	-0.383	0.0
32	-0.171	-0.024

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.100E 01 NWIND=999
NORM(X)= 0.283E 01 NORM(G)= 0.283E 01 NORM(E)= 0.527E-04

I/J	1	2
1	0.0	0.0
2	0.195	0.000
3	0.383	0.0
4	0.556	0.000
5	0.707	0.0
6	0.831	0.000
7	0.924	0.0
8	0.981	0.000
9	1.000	0.0
10	0.981	0.000
11	0.924	0.0
12	0.831	0.000
13	0.707	0.0
14	0.556	0.000
15	0.383	0.0
16	0.195	0.000
17	0.000	0.0
18	-0.195	-0.000
19	-0.383	0.0
20	-0.556	-0.000
21	-0.707	0.0
22	-0.831	-0.000
23	-0.924	0.0
24	-0.981	-0.000
25	-1.000	0.0
26	-0.981	-0.000
27	-0.924	0.0
28	-0.831	-0.000
29	-0.707	0.0
30	-0.556	-0.000
31	-0.383	0.0
32	-0.195	-0.000

PRINT OF FFT INTERPOLATION
MULTIPLIER= 0.100E 01 NWIND=999
NORM(X)= 0.283E 01 NORM(G)= 0.283E 01 NORM(E)= 0.453E-05

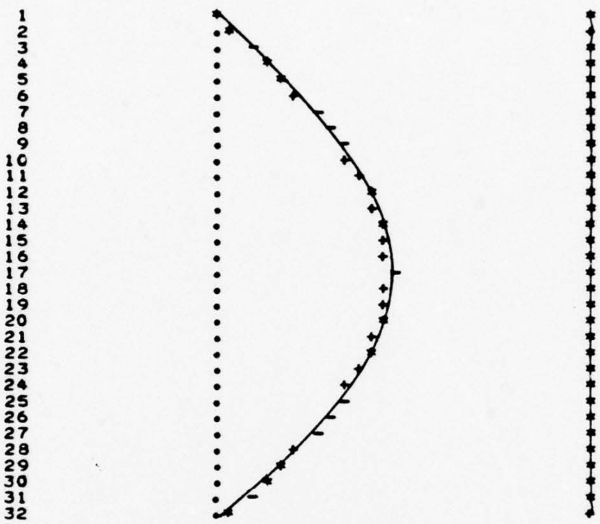
I/J	1	2
1	0.0	0.0
2	0.195	-0.000
3	0.383	0.0
4	0.556	-0.000
5	0.707	0.0
6	0.831	0.000
7	0.924	0.0
8	0.981	0.000
9	1.000	0.0
10	0.981	0.000
11	0.924	0.0
12	0.831	0.000
13	0.707	0.0
14	0.556	0.000
15	0.383	0.0
16	0.195	0.000
17	0.000	0.0
18	-0.195	0.000
19	-0.383	0.0
20	-0.556	-0.000
21	-0.707	0.0
22	-0.831	-0.000
23	-0.924	0.0
24	-0.981	-0.000
25	-1.000	0.0
26	-0.981	-0.000
27	-0.924	0.0
28	-0.831	-0.000
29	-0.707	0.0
30	-0.556	-0.000
31	-0.383	0.0
32	-0.195	-0.000

8-1c Tabular Comparison of Interpolation Schemes - Signal 1

PLOT OF MATRIX INTERPOLATION
MULTIPLIER= 0.100E 01 NWIND= 0
EIGENVECTORS USED IN THE RECONSTRUCTION
1 2 3 4 5 6 7 8

9 1C 11 12 13 14 15 16

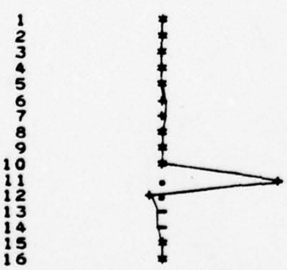
COLUMN 1 IS INTERPCLATED CATA
COLUMN 2 IS THE EFFOR
CCLUMNS FLOTTED. 1. 2.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27

PLOT OF CORRELATICN COEFFICIENTS
MULTIPLIER= 0.100E 01 NWIND= 0
CCLUMNS FLOTTED. 1.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

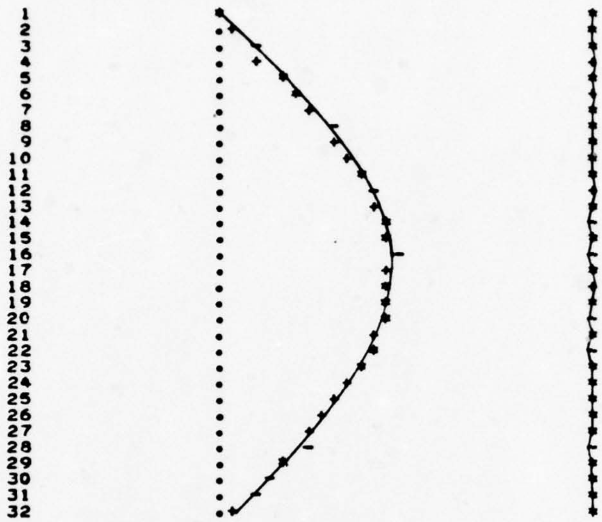
NUMBER OF PEAK-TO-PEAK PRINT CELLS= 19

8-2a Transient Interpolation and Correlation Coefficients - Signal 2

PLOT OF MATRIX INTERPOLATION
MULTIPLIER= 0.101E 01 NWIND= 0
EIGENVECTORS USED IN THE RECONSTRUCTION
0 0 0 0 0 0 0 0

0 0 11 0 0 0 0 0

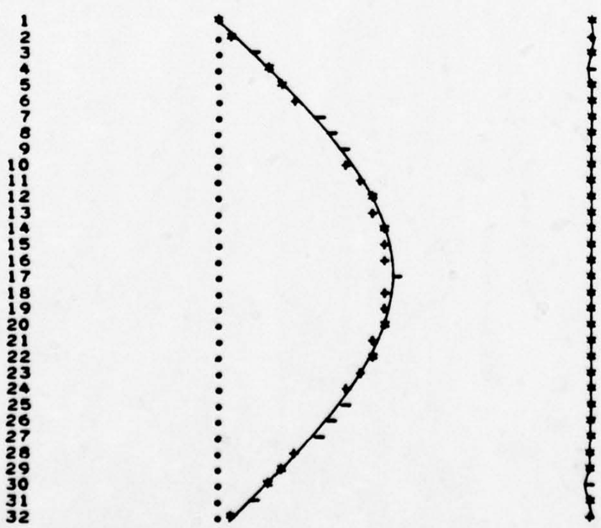
COLUMN 1 IS INTERPOLATED DATA
COLUMN 2 IS THE ERROR
COLUMNS PLOTTED, 1. 2.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27

PLOT OF FFT INTERPOLATION
MULTIPLIER= 0.100E 01 NWIND=999
COLUMN 1 IS INTERPOLATED DATA
COLUMN 2 IS THE ERROR
COLUMNS PLOTTED, 1. 2.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27

8-2b Eigenfilter and FFT Interpolation - Signal 2

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.100E 01 NWIND= 0
NORM(X)= 0.283E 01 NORM(G)= 0.283E 01 NORM(E)= 0.277E-01

I/J	1	2
1	0.0	0.0
2	0.081	0.017
3	0.195	0.0
4	0.298	-0.008
5	0.383	0.0
6	0.467	0.005
7	0.556	0.0
8	0.637	-0.003
9	0.707	0.0
10	0.771	0.002
11	0.831	0.0
12	0.883	-0.001
13	0.924	0.0
14	0.956	0.001
15	0.981	0.0
16	0.995	-0.000
17	1.000	0.0
18	0.995	-0.000
19	0.981	0.0
20	0.956	0.001
21	0.924	0.0
22	0.883	-0.001
23	0.831	0.0
24	0.771	0.002
25	0.707	0.0
26	0.637	-0.003
27	0.556	0.0
28	0.467	0.005
29	0.383	0.0
30	0.298	-0.008
31	0.195	0.0
32	0.081	0.017

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.101E 01 NWIND= 0
NORM(X)= 0.283E 01 NORM(G)= 0.283E 01 NORM(E)= 0.680E-01

I/J	1	2
1	0.0	0.0
2	0.088	0.009
3	0.193	0.0
4	0.258	0.029
5	0.378	0.0
6	0.437	0.029
7	0.549	0.0
8	0.636	-0.009
9	0.699	0.0
10	0.755	0.009
11	0.822	0.0
12	0.855	0.018
13	0.914	0.0
14	0.960	-0.014
15	0.970	0.0
16	1.000	-0.016
17	0.985	0.0
18	0.971	0.013
19	0.970	0.0
20	0.961	-0.015
21	0.914	0.0
22	0.896	-0.024
23	0.822	0.0
24	0.760	0.004
25	0.699	0.0
26	0.626	0.001
27	0.549	0.0
28	0.493	-0.027
29	0.378	0.0
30	0.279	0.008
31	0.193	0.0
32	0.087	0.010

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.100E 01 NWIND=999
NORM(X)= 0.283E 01 NORM(G)= 0.283E 01 NORM(E)= 0.595E-01

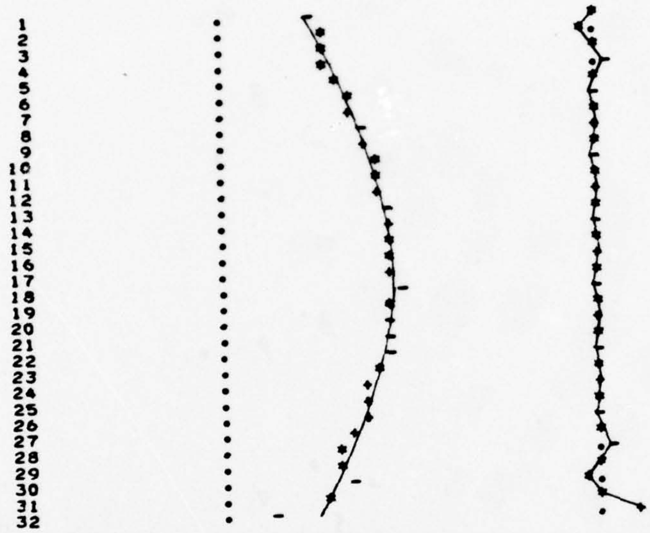
I/J	1	2
1	0.0	0.0
2	0.063	0.036
3	0.195	0.0
4	0.308	-0.017
5	0.383	0.0
6	0.461	0.011
7	0.556	0.0
8	0.642	-0.007
9	0.707	0.0
10	0.768	0.005
11	0.831	0.0
12	0.885	-0.003
13	0.924	0.0
14	0.955	0.002
15	0.981	0.0
16	0.996	-0.001
17	1.000	0.0
18	0.996	-0.001
19	0.981	0.0
20	0.955	0.002
21	0.924	0.0
22	0.885	-0.003
23	0.831	0.0
24	0.768	0.005
25	0.707	0.0
26	0.642	-0.007
27	0.556	0.0
28	0.461	0.011
29	0.383	0.0
30	0.308	-0.017
31	0.195	0.0
32	0.063	0.036

PRINT OF FFT INTERPOLATION
MULTIPLIER= 0.100E 01 NWIND=999
NORM(X)= 0.283E 01 NORM(G)= 0.283E 01 NORM(E)= 0.595E-01

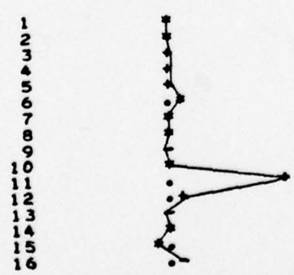
I/J	1	2
1	0.0	0.0
2	0.063	0.036
3	0.195	0.0
4	0.308	-0.017
5	0.383	0.0
6	0.461	0.011
7	0.556	0.0
8	0.642	-0.007
9	0.707	0.0
10	0.768	0.005
11	0.831	0.0
12	0.885	-0.003
13	0.924	0.0
14	0.955	0.002
15	0.981	0.0
16	0.996	-0.001
17	1.000	0.0
18	0.996	-0.001
19	0.981	0.0
20	0.955	0.002
21	0.924	0.0
22	0.885	-0.003
23	0.831	0.0
24	0.768	0.005
25	0.707	0.0
26	0.642	-0.007
27	0.556	0.0
28	0.461	0.011
29	0.383	0.0
30	0.308	-0.017
31	0.195	0.0
32	0.063	0.036

8-2c Tabular Comparison of Interpolation Schemes - Signal 2

```
PLOT OF MATRIX INTERPOLATION
MULTIPLIER= 0.102E 01 NWIND=  0
EIGENVECTORS USED IN THE RECONSTRUCTION
  1  2  3  4  5  6  7  8

  9 10 11 12 13 14 15 16

COLUMN 1 IS INTERPOLATED DATA
COLUMN 2 IS THE ERROR
COLUMNS PLOTTED.  1.  2.
```



```
LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27


PLOT OF CORRELATION COEFFICIENTS
MULTIPLIER= 0.100E 01 NWIND=  0
COLUMNS PLOTTED.  1.
```



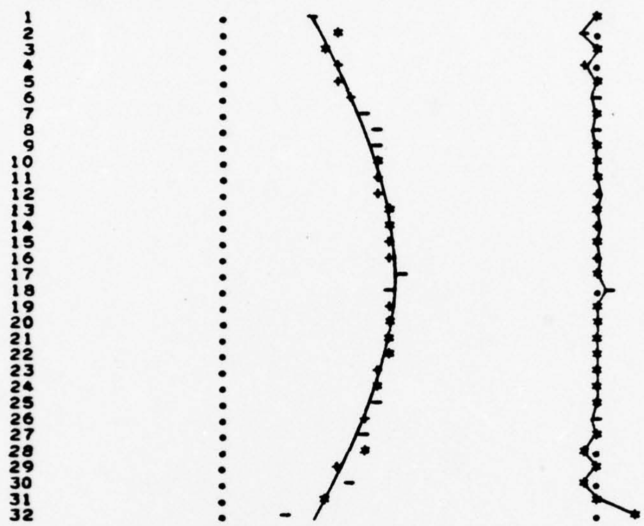
```
LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 19
```

8-3a Transient Interpolation and Correlation Coefficients - Signal 3

PLOT OF MATRIX INTERPOLATION
MULTIPLIER= 0.100E 01 NWIND= 0
EIGENVECTORS USED IN THE RECONSTRUCTION
0 0 0 0 0 0 0 0

0 0 11 0 0 0 0 0

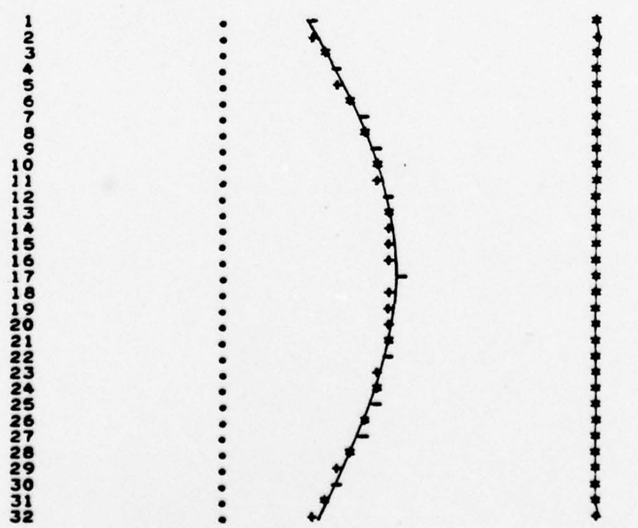
COLUMN 1 IS INTERPCLATED CATA
COLUMN 2 IS THE ERROR
COLUMNS FLOTTED. 1. 2.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER CF PEAK-TO-PEAK PRINT CELLS= 27

PLOT OF FFT INTERPOLATION
MULTIPLIER= 0.100E 01 NWIND=999
COLUMN 1 IS INTERPCLATED DATA
COLUMN 2 IS THE ERROR
COLUMNS FLOTTED. 1. 2.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27

8-3b Eigenfilter and FFT Interpolation - Signal 3

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.102E 01 NWIND= 0
NORM(X)= 0.333E 01 NORM(G)= 0.332E 01 NORM(E)= 0.300E 00

I/J	1	2
1	0.491	0.0
2	0.603	-0.064
3	0.587	0.0
4	0.595	0.039
5	0.679	0.0
6	0.752	-0.029
7	0.764	0.0
8	0.779	0.024
9	0.839	0.0
10	0.893	-0.022
11	0.900	0.0
12	0.904	0.020
13	0.945	0.0
14	0.981	-0.019
15	0.973	0.0
16	0.961	0.019
17	0.982	0.0
18	1.000	-0.020
19	0.973	0.0
20	0.940	0.021
21	0.945	0.0
22	0.948	-0.023
23	0.900	0.0
24	0.844	0.027
25	0.839	0.0
26	0.836	-0.033
27	0.764	0.0
28	0.678	0.045
29	0.679	0.0
30	0.710	-0.076
31	0.587	0.0
32	0.281	0.259

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.100E 01 NWIND= 0
NORM(X)= 0.333E 01 NORM(G)= 0.335E 01 NORM(E)= 0.281E 00

I/J	1	2
1	0.500	0.0
2	0.658	-0.109
3	0.598	0.0
4	0.697	-0.052
5	0.691	0.0
6	0.757	-0.021
7	0.778	0.0
8	0.854	-0.037
9	0.854	0.0
10	0.878	0.008
11	0.916	0.0
12	0.904	0.037
13	0.962	0.0
14	0.965	0.014
15	0.990	0.0
16	0.980	0.018
17	1.000	0.0
18	0.947	0.051
19	0.990	0.0
20	0.966	0.012
21	0.962	0.0
22	0.952	-0.011
23	0.916	0.0
24	0.884	0.002
25	0.854	0.0
26	0.842	-0.025
27	0.778	0.0
28	0.821	-0.086
29	0.691	0.0
30	0.722	-0.077
31	0.598	0.0
32	0.339	0.210

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.100E 01 NWIND=999
NORM(X)= 0.333E 01 NORM(G)= 0.333E 01 NORM(E)= 0.297E-01

I/J	1	2
1	0.500	0.0
2	0.531	0.018
3	0.598	0.0
4	0.654	-0.009
5	0.691	0.0
6	0.730	0.005
7	0.778	0.0
8	0.821	-0.004
9	0.854	0.0
10	0.884	0.002
11	0.916	0.0
12	0.943	-0.002
13	0.962	0.0
14	0.978	0.001
15	0.990	0.0
16	0.998	-0.000
17	1.000	0.0
18	0.998	-0.000
19	0.990	0.0
20	0.978	0.001
21	0.962	0.0
22	0.943	-0.002
23	0.916	0.0
24	0.884	0.002
25	0.854	0.0
26	0.821	-0.004
27	0.778	0.0
28	0.730	0.005
29	0.691	0.0
30	0.654	-0.009
31	0.598	0.0
32	0.531	0.018

PRINT OF FFT INTERPOLATION
MULTIPLIER= 0.100E 01 NWIND=999
NORM(X)= 0.333E 01 NORM(G)= 0.333E 01 NORM(E)= 0.297E-01

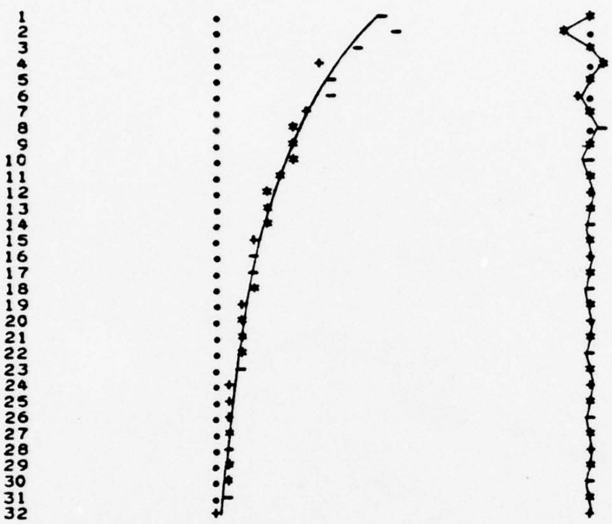
I/J	1	2
1	0.500	0.0
2	0.531	0.018
3	0.598	0.0
4	0.654	-0.009
5	0.691	0.0
6	0.730	0.005
7	0.778	0.0
8	0.821	-0.004
9	0.854	0.0
10	0.884	0.002
11	0.916	0.0
12	0.943	-0.002
13	0.962	0.0
14	0.978	0.001
15	0.990	0.0
16	0.998	-0.000
17	1.000	0.0
18	0.998	-0.000
19	0.990	0.0
20	0.978	0.001
21	0.962	0.0
22	0.943	-0.002
23	0.916	0.0
24	0.884	0.002
25	0.854	0.0
26	0.821	-0.004
27	0.778	0.0
28	0.730	0.005
29	0.691	0.0
30	0.654	-0.009
31	0.598	0.0
32	0.531	0.018

8-3c Tabular Comparison of Interpolation Schemes - Signal 3

PLCT CF MATRIX INTERPOLATICN
MULTIPLIER= 0.107E 01 NWIND= 0
EIGENVECTORS USED IN THE RECONSTRUCTION
1 2 3 4 5 6 7 8

9 10 11 12 13 14 15 16

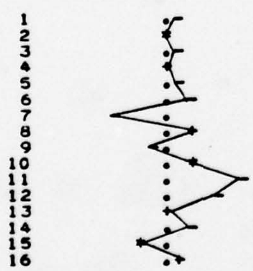
CCLUMN 1 IS INTERPCLATED CATA
CCLUMN 2 IS THE ERROR
COLUMNS FLOTTED. 1. 2.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER CF PEAK-TO-PEAK PRINT CELLS= 27

PLOT OF CORRELATICN COEFFICIENTS
MULTIPLIER= 0.100E 01 NWIND= 0
COLUMNS FLOTTED. 1.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

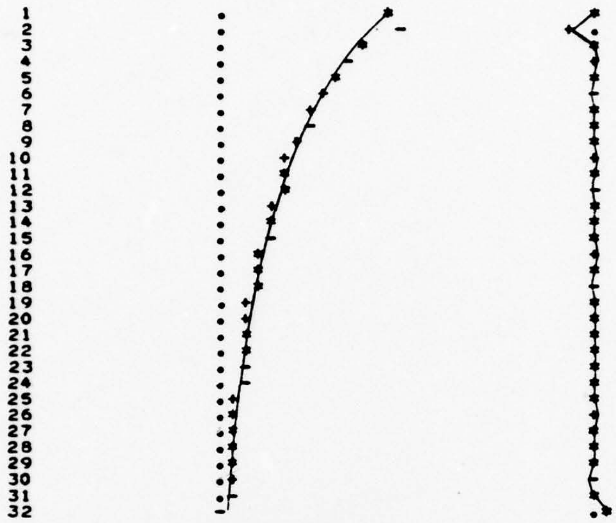
NUMBER CF PEAK-TO-FEAK PRINT CELLS= 19

8-4a Transient Interpolation and Correlation Coefficients - Signal 4

PLOT OF MATRIX INTERPOLATION
MULTIPLIER= 0.103E 01 NWIND= 0
EIGENVECTORS USED IN THE RECONSTRUCTION
0 0 0 0 0 6 7 8

0 10 11 12 0 14 15 0

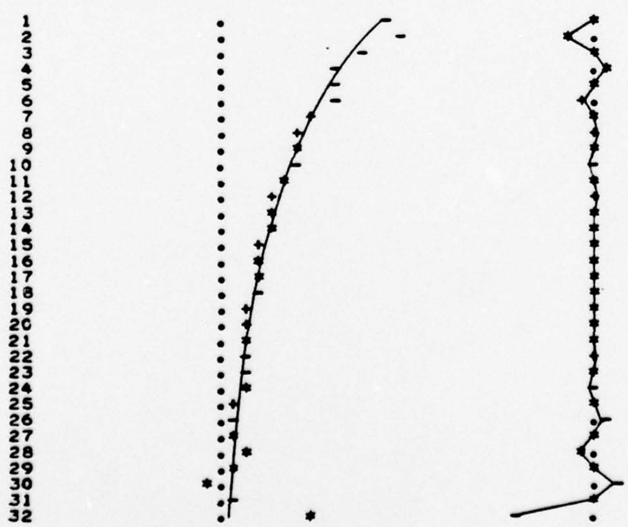
COLUMN 1 IS INTERPOLATED DATA
COLUMN 2 IS THE ERROR
COLUMNS PLOTTED, 1, 2,



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27

PLOT OF FFT INTERPOLATION
MULTIPLIER= 0.106E 01 NWIND=999
COLUMN 1 IS INTERPOLATED DATA
COLUMN 2 IS THE ERROR
COLUMNS PLOTTED, 1, 2,



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27

8-4b Eigenfilter and FFT Interpolation - Signal 4

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.107E 01 NWIND= 0
NORM(X)= 0.179E 01 NORM(G)= 0.170E 01 NORM(E)= 0.208E 00

I/J	1	2
1	0.939	0.0
2	1.000	-0.145
3	0.778	0.0
4	0.629	0.080
5	0.645	0.0
6	0.642	-0.055
7	0.535	0.0
8	0.446	0.041
9	0.443	0.0
10	0.437	-0.033
11	0.368	0.0
12	0.307	0.028
13	0.305	0.0
14	0.302	-0.024
15	0.253	0.0
16	0.209	0.021
17	0.209	0.0
18	0.210	-0.019
19	0.174	0.0
20	0.140	0.018
21	0.144	0.0
22	0.148	-0.016
23	0.119	0.0
24	0.093	0.016
25	0.099	0.0
26	0.105	-0.015
27	0.082	0.0
28	0.059	0.015
29	0.068	0.0
30	0.080	-0.018
31	0.056	0.0
32	0.017	0.034

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.103E 01 NWIND= 0
NORM(X)= 0.179E 01 NORM(G)= 0.169E 01 NORM(E)= 0.150E 00

I/J	1	2
1	0.974	0.0
2	1.000	-0.113
3	0.807	0.0
4	0.717	0.019
5	0.669	0.0
6	0.622	-0.013
7	0.555	0.0
8	0.498	0.007
9	0.460	0.0
10	0.402	0.017
11	0.381	0.0
12	0.362	-0.015
13	0.316	0.0
14	0.287	0.001
15	0.262	0.0
16	0.221	0.018
17	0.217	0.0
18	0.219	-0.021
19	0.180	0.0
20	0.165	-0.001
21	0.149	0.0
22	0.138	-0.002
23	0.124	0.0
24	0.112	0.001
25	0.103	0.0
26	0.076	0.017
27	0.085	0.0
28	0.081	-0.004
29	0.071	0.0
30	0.089	-0.025
31	0.058	0.0
32	-0.023	0.076

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.106E 01 NWIND=999
NORM(X)= 0.179E 01 NORM(G)= 0.178E 01 NORM(E)= 0.547E 00

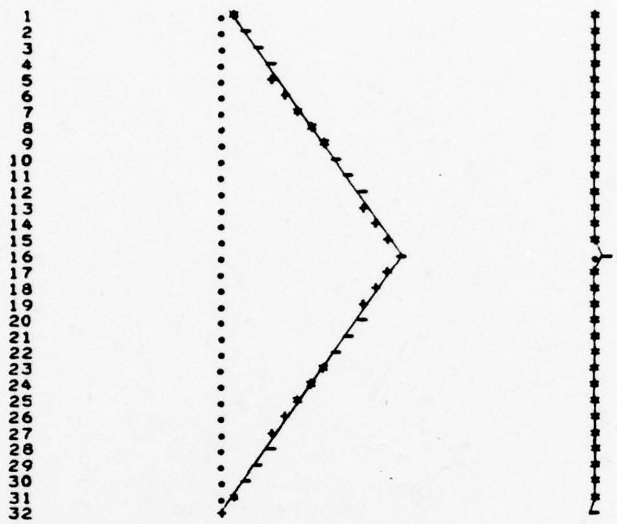
I/J	1	2
1	0.948	0.0
2	1.000	-0.137
3	0.786	0.0
4	0.643	0.072
5	0.651	0.0
6	0.639	-0.046
7	0.540	0.0
8	0.461	0.031
9	0.448	0.0
10	0.428	-0.021
11	0.371	0.0
12	0.325	0.013
13	0.308	0.0
14	0.287	-0.006
15	0.255	0.0
16	0.232	0.000
17	0.211	0.0
18	0.187	0.006
19	0.175	0.0
20	0.172	-0.012
21	0.145	0.0
22	0.112	0.020
23	0.121	0.0
24	0.139	-0.030
25	0.100	0.0
26	0.047	0.044
27	0.083	0.0
28	0.144	-0.069
29	0.069	0.0
30	-0.066	0.128
31	0.057	0.0
32	0.517	-0.465

PRINT OF FFT INTERPOLATION
MULTIPLIER= 0.106E 01 NWIND=999
NORM(X)= 0.179E 01 NORM(G)= 0.178E 01 NORM(E)= 0.547E 00

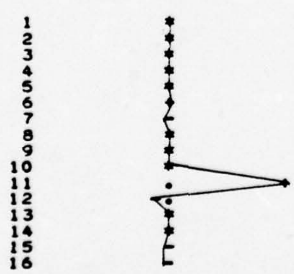
I/J	1	2
1	0.948	0.0
2	1.000	-0.137
3	0.786	0.0
4	0.643	0.072
5	0.651	0.0
6	0.639	-0.046
7	0.540	0.0
8	0.461	0.031
9	0.448	0.0
10	0.428	-0.021
11	0.371	0.0
12	0.325	0.013
13	0.308	0.0
14	0.287	-0.006
15	0.255	0.0
16	0.232	0.000
17	0.211	0.0
18	0.187	0.006
19	0.175	0.0
20	0.172	-0.012
21	0.145	0.0
22	0.112	0.020
23	0.121	0.0
24	0.139	-0.030
25	0.100	0.0
26	0.047	0.044
27	0.083	0.0
28	0.144	-0.069
29	0.069	0.0
30	-0.066	0.128
31	0.057	0.0
32	0.517	-0.465

8-4c Tabular Comparison of Interpolation Schemes - Signal 4

```
PLOT OF MATRIX INTERPOLATION
MULTIPLIER= 0.960E 00 NWIND=  0
EIGENVECTORS USED IN THE RECONSTRUCTION
  1  2  3  4  5  6  7  8

  9 10 11 12 13 14 15 16

COLUMN 1 IS INTERPOLATED DATA
COLUMN 2 IS THE ERROR
COLUMNS PLOTTED.  1.  2.
```



```
LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27


PLOT OF CORRELATION COEFFICIENTS
MULTIPLIER= 0.100E 01 NWIND=  0
COLUMNS PLOTTED.  1.
```



```
LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 19
```

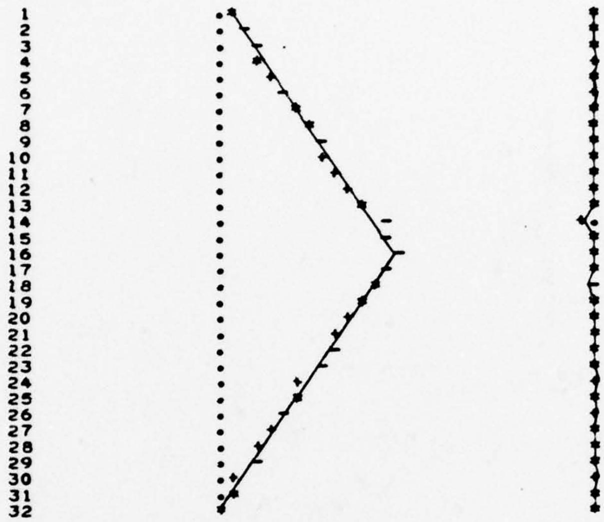
8-5a Transient Interpolation and Correlation Coefficients - Signal 5

PLOT OF MATRIX INTERPOLATION
MULTIPLIER= 0.992E 00 NWIND= 0
EIGENVECTORS USED IN THE RECONSTRUCTION
0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 11 0 0 0 0 0

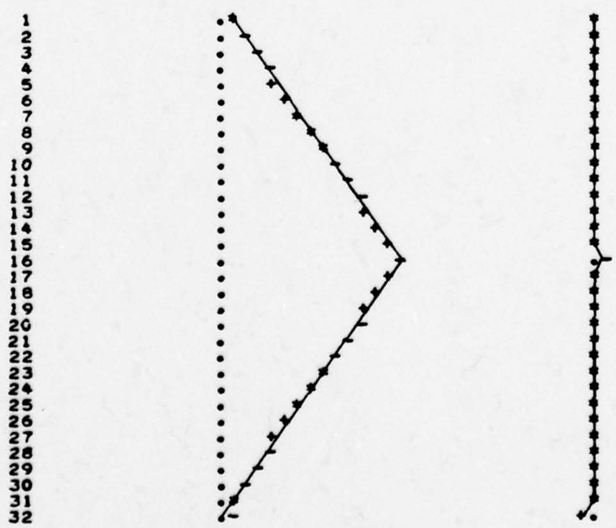
COLUMN 1 IS INTERPOLATED DATA
COLUMN 2 IS THE ERROR
COLUMNS PLOTTED, 1. 2.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27

PLOT OF FFT INTERPOLATION
MULTIPLIER= 0.960E 00 NWIND=999
COLUMN 1 IS INTERPOLATED DATA
COLUMN 2 IS THE ERROR
COLUMNS PLOTTED, 1. 2.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27

8-5b Eigenfilter and FFT Interpolation - Signal 5

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.960E 00 NWIND= 0
NORM(X)= 0.230E 01 NORM(G)= 0.230E 01 NORM(E)= 0.452E-01

I/J	1	2
1	0.065	0.0
2	0.127	0.003
3	0.195	0.0
4	0.261	-0.001
5	0.325	0.0
6	0.390	0.000
7	0.456	0.0
8	0.520	0.000
9	0.586	0.0
10	0.652	-0.001
11	0.716	0.0
12	0.779	0.002
13	0.846	0.0
14	0.917	-0.006
15	0.976	0.0
16	1.000	0.041
17	0.976	0.0
18	0.917	-0.006
19	0.846	0.0
20	0.779	0.002
21	0.716	0.0
22	0.652	-0.001
23	0.586	0.0
24	0.520	0.000
25	0.456	0.0
26	0.390	0.000
27	0.325	0.0
28	0.261	-0.001
29	0.195	0.0
30	0.127	0.003
31	0.065	0.0
32	0.021	-0.021

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.992E 00 NWIND= 0
NORM(X)= 0.230E 01 NORM(G)= 0.232E 01 NORM(E)= 0.843E-01

I/J	1	2
1	0.063	0.0
2	0.132	-0.006
3	0.189	0.0
4	0.234	0.018
5	0.315	0.0
6	0.355	0.023
7	0.441	0.0
8	0.511	-0.007
9	0.567	0.0
10	0.627	0.004
11	0.693	0.0
12	0.756	-0.000
13	0.819	0.0
14	0.926	-0.044
15	0.945	0.0
16	1.000	0.008
17	0.945	0.0
18	0.856	-0.014
19	0.819	0.0
20	0.767	-0.010
21	0.693	0.0
22	0.630	0.000
23	0.567	0.0
24	0.470	0.034
25	0.441	0.0
26	0.342	0.036
27	0.315	0.0
28	0.240	0.012
29	0.189	0.0
30	0.091	0.035
31	0.063	0.0
32	0.011	-0.011

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.960E 00 NWIND=999
NORM(X)= 0.230E 01 NORM(G)= 0.230E 01 NORM(E)= 0.571E-01

I/J	1	2
1	0.065	0.0
2	0.125	0.006
3	0.195	0.0
4	0.262	-0.002
5	0.325	0.0
6	0.390	0.001
7	0.456	0.0
8	0.521	0.000
9	0.586	0.0
10	0.651	-0.001
11	0.716	0.0
12	0.779	0.002
13	0.846	0.0
14	0.917	-0.006
15	0.976	0.0
16	1.000	0.041
17	0.976	0.0
18	0.917	-0.006
19	0.846	0.0
20	0.779	0.002
21	0.716	0.0
22	0.651	-0.001
23	0.586	0.0
24	0.521	0.000
25	0.456	0.0
26	0.390	0.001
27	0.325	0.0
28	0.262	-0.002
29	0.195	0.0
30	0.125	0.006
31	0.065	0.0
32	0.041	-0.041

PRINT OF FFT INTERPOLATION
MULTIPLIER= 0.960E 00 NWIND=999
NORM(X)= 0.230E 01 NORM(G)= 0.230E 01 NORM(E)= 0.571E-01

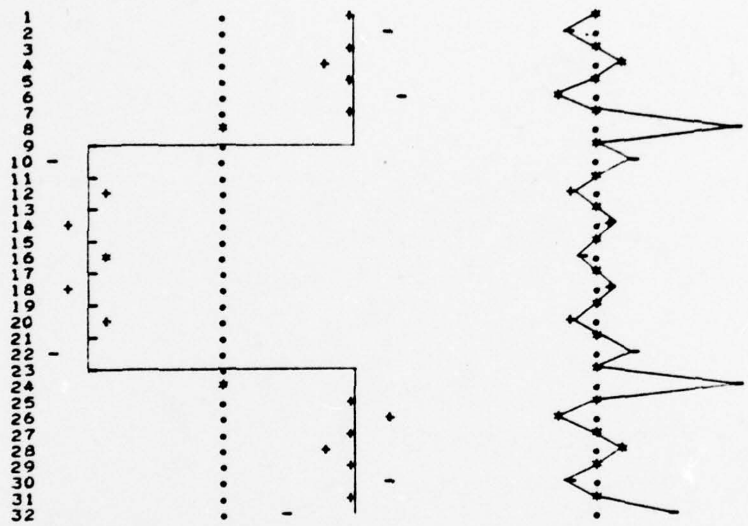
I/J	1	2
1	0.065	0.0
2	0.125	0.006
3	0.195	0.0
4	0.262	-0.002
5	0.325	0.0
6	0.390	0.001
7	0.456	0.0
8	0.521	-0.000
9	0.586	0.0
10	0.651	-0.001
11	0.716	0.0
12	0.779	0.002
13	0.846	0.0
14	0.917	-0.006
15	0.976	0.0
16	1.000	0.041
17	0.976	0.0
18	0.917	-0.006
19	0.846	0.0
20	0.779	0.002
21	0.716	0.0
22	0.651	-0.001
23	0.586	0.0
24	0.521	-0.000
25	0.456	0.0
26	0.390	0.001
27	0.325	0.0
28	0.262	-0.002
29	0.195	0.0
30	0.125	0.006
31	0.065	0.0
32	0.041	-0.041

8-5c Tabular Comparison of Interpolation Schemes - Signal 5

PLOT OF MATRIX INTERPOLATION
MULTIPLIER= 0.130E 01 NWIND= 0
EIGENVECTORS USED IN THE RECONSTRUCTION
1 2 3 4 5 6 7 8

9 10 11 12 13 14 15 16

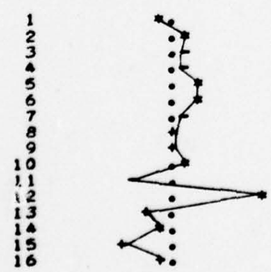
COLUMN 1 IS INTERPOLATED DATA
COLUMN 2 IS THE ERROR
COLUMNS PLOTTED, 1. 2.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27

PLOT OF CORRELATION COEFFICIENTS
MULTIPLIER= 0.100E 01 NWIND= 0
COLUMNS PLOTTED, 1.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

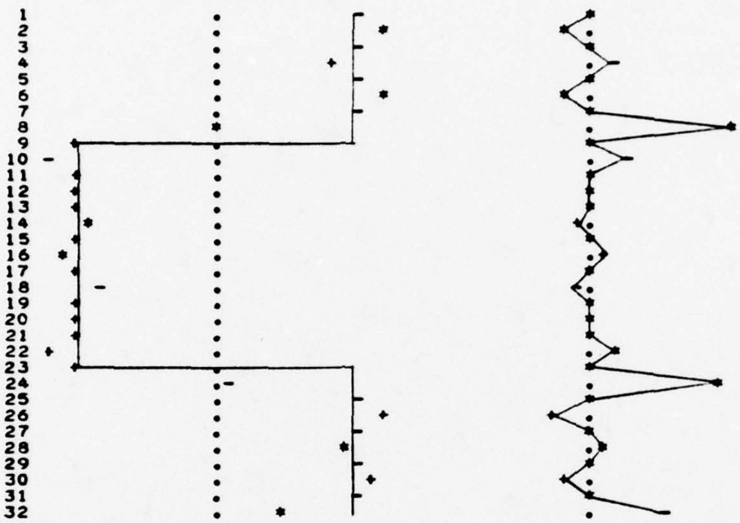
NUMBER OF PEAK-TO-PEAK PRINT CELLS= 19

8-6a Transient Interpolation and Correlation Coefficients - Signal 6

PLOT OF MATRIX INTERPOLATICN
MULTIPLIER= 0.126E 01 NWIND= 0
EIGENVECTORS USED IN THE RECONSTRUCTICN
0 0 0 0 5 6 0 0

0 0 11 12 13 0 15 0

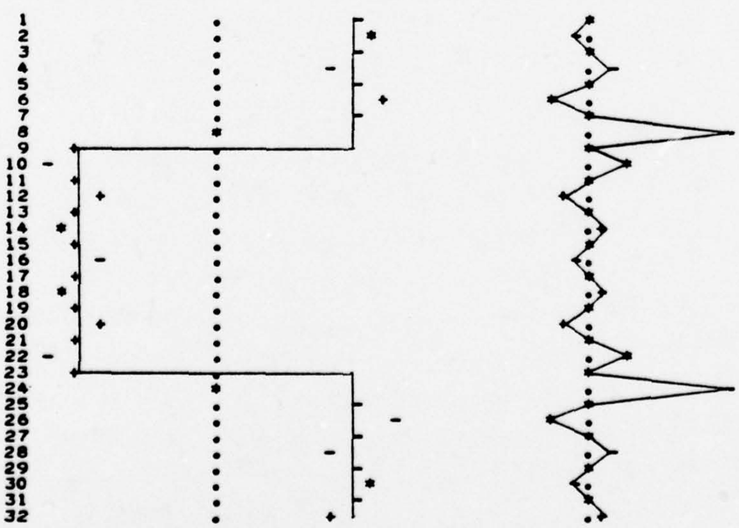
COLUMN 1 IS INTERPCLATED CATA
COLUMN 2 IS THE ERRCR
COLUMNS PLOTTED, 1, 2,



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27

PLOT OF FFT INTERPCLATICN
MULTIPLIER= 0.128E 01 NWIND=959
COLUMN 1 IS INTERPCLATED CATA
COLUMN 2 IS THE ERRCR
COLUMNS PLOTTED, 1, 2,



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER CF PEAK-TO-PEAK PRINT CELLS= 27

8-6b Eigenfilter and FFT Interpolation - Signal 6

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.130E 01 NWIND= 0
NORM(X)= 0.400E 01 NORM(G)= 0.397E 01 NORM(E)= 0.173E 01

I/J	1	2
1	0.768	0.0
2	0.938	-0.170
3	0.768	0.0
4	0.610	0.159
5	0.768	0.0
6	1.000	-0.232
7	0.768	0.0
8	-0.010	0.778
9	-0.768	0.0
10	-0.977	0.209
11	-0.768	0.0
12	-0.644	-0.124
13	-0.768	0.0
14	-0.865	0.097
15	-0.768	0.0
16	-0.678	-0.090
17	-0.768	0.0
18	-0.865	0.097
19	-0.768	0.0
20	-0.644	-0.124
21	-0.768	0.0
22	-0.977	0.209
23	-0.768	0.0
24	-0.010	0.778
25	0.768	0.0
26	1.000	-0.212
27	0.768	0.0
28	0.610	0.159
29	0.768	0.0
30	0.938	-0.170
31	0.768	0.0
32	0.337	0.432

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.126E 01 NWIND= 0
NORM(X)= 0.400E 01 NORM(G)= 0.393E 01 NORM(E)= 0.158E 01

I/J	1	2
1	0.795	0.0
2	0.952	-0.157
3	0.795	0.0
4	0.682	0.113
5	0.795	0.0
6	0.953	-0.158
7	0.795	0.0
8	-0.012	0.807
9	-0.795	0.0
10	-1.000	0.205
11	-0.795	0.0
12	-0.793	-0.002
13	-0.795	0.0
14	-0.736	-0.059
15	-0.795	0.0
16	-0.901	0.106
17	-0.795	0.0
18	-0.702	-0.093
19	-0.795	0.0
20	-0.784	-0.011
21	-0.795	0.0
22	-0.950	0.155
23	-0.795	0.0
24	0.053	0.742
25	0.795	0.0
26	0.986	-0.191
27	0.795	0.0
28	0.730	0.065
29	0.795	0.0
30	0.906	-0.111
31	0.795	0.0
32	0.381	0.414

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.128E 01 NWIND=999
NORM(X)= 0.400E 01 NORM(G)= 0.400E 01 NORM(E)= 0.159E 01

I/J	1	2
1	0.780	0.0
2	0.883	-0.103
3	0.780	0.0
4	0.649	0.132
5	0.780	0.0
6	1.000	-0.220
7	0.780	0.0
8	0.000	0.780
9	-0.780	0.0
10	-1.000	0.220
11	-0.780	0.0
12	-0.649	-0.132
13	-0.780	0.0
14	-0.883	0.103
15	-0.780	0.0
16	-0.665	-0.096
17	-0.780	0.0
18	-0.883	0.103
19	-0.780	0.0
20	-0.649	-0.132
21	-0.780	0.0
22	-1.000	0.220
23	-0.780	0.0
24	0.000	0.780
25	0.780	0.0
26	1.000	-0.220
27	0.780	0.0
28	0.649	0.132
29	0.780	0.0
30	0.883	-0.103
31	0.780	0.0
32	0.665	0.096

PRINT OF FFT INTERPOLATION
MULTIPLIER= 0.128E 01 NWIND=999
NORM(X)= 0.400E 01 NORM(G)= 0.400E 01 NORM(E)= 0.159E 01

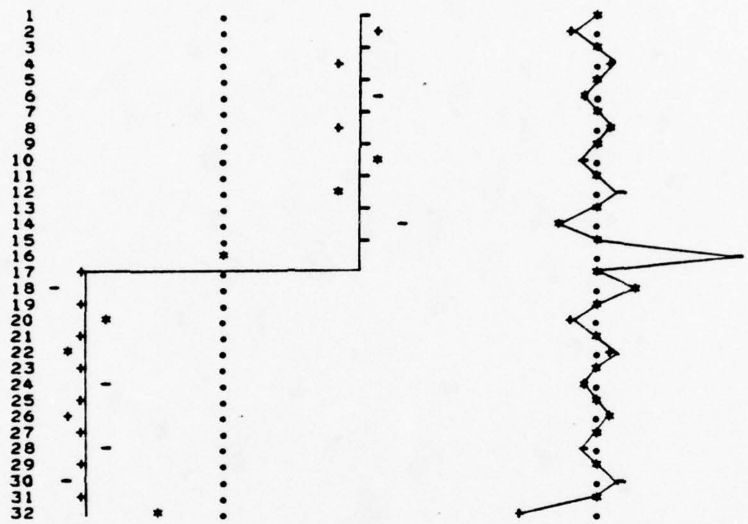
I/J	1	2
1	0.780	0.0
2	0.883	-0.103
3	0.780	0.0
4	0.649	0.132
5	0.780	0.0
6	1.000	-0.220
7	0.780	0.0
8	-0.000	0.780
9	-0.780	0.0
10	-1.000	0.220
11	-0.780	0.0
12	-0.649	-0.132
13	-0.780	0.0
14	-0.883	0.103
15	-0.780	0.0
16	-0.665	-0.096
17	-0.780	0.0
18	-0.883	0.103
19	-0.780	0.0
20	-0.649	-0.132
21	-0.780	0.0
22	-1.000	0.220
23	-0.780	0.0
24	0.000	0.780
25	0.780	0.0
26	1.000	-0.220
27	0.780	0.0
28	0.649	0.132
29	0.780	0.0
30	0.883	-0.103
31	0.780	0.0
32	0.665	0.096

8-6c Tabular Comparison of Interpolation Schemes - Signal 6

PLOT OF MATRIX INTERPOLATION
MULTIPLIER= 0.128E 01 NWIND= 0
EIGENVECTORS USED IN THE RECONSTRUCTICN
1 2 3 4 5 6 7 8

9 10 11 12 13 14 15 16

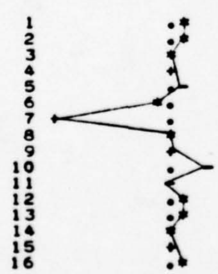
COLUMN 1 IS INTERPOLATED DATA
COLUMN 2 IS THE ERRCR
COLUMNS FLOTTED. 1. 2.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER CF PEAK-TO-PEAK PRINT CELLS= 27

PLOT OF CORRELATICN COEFFICIENTS
MULTIPLIER= 0.100E 01 NWIND= 0
COLUMNS FLOTTED. 1.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

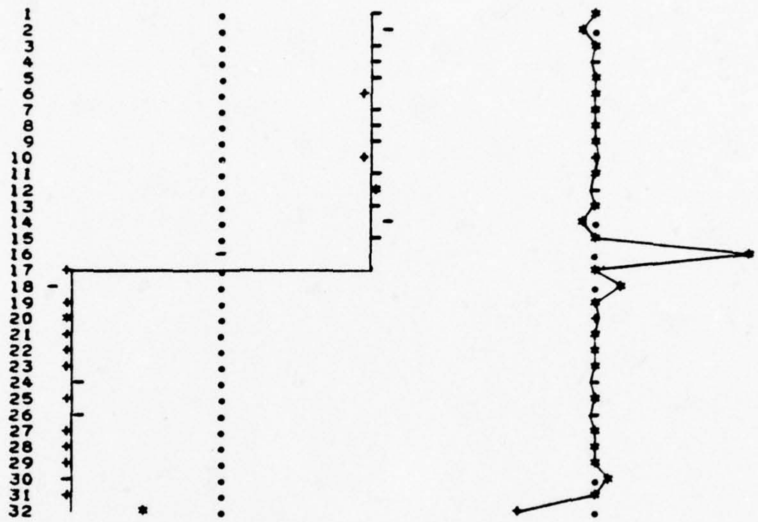
NUMBER CF PEAK-TO-FEAK PRINT CELLS= 19

8-7a Transient Interpolation and Correlation Coefficients - Signal 7

PLOT OF MATRIX INTERPOLATION
MULTIPLIER= 0.117E 01 NWIND= 0
EIGENVECTORS USED IN THE RECONSTRUCTION
0 0 0 0 0 0 7 0

0 10 0 0 0 0 0 0

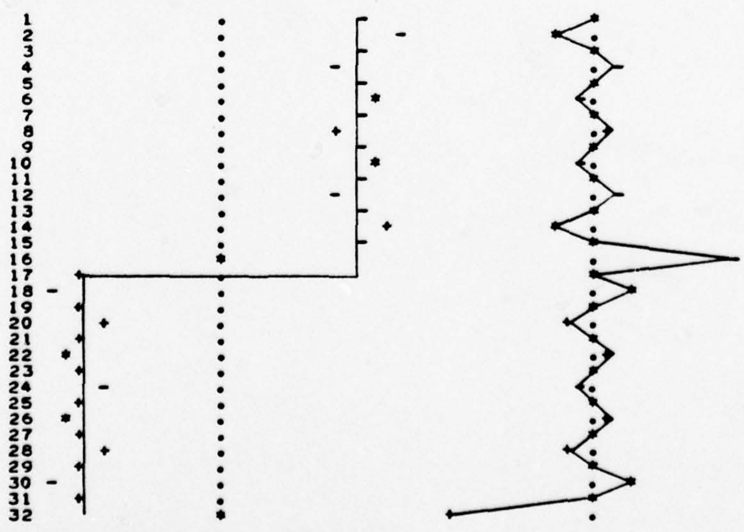
COLUMN 1 IS INTERPOLATED DATA
COLUMN 2 IS THE ERROR
COLUMNS PLOTTED, 1, 2,



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27

PLOT OF FFT INTERPOLATION
MULTIPLIER= 0.128E 01 NWIND=999
COLUMN 1 IS INTERPOLATED DATA
COLUMN 2 IS THE ERROR
COLUMNS PLOTTED, 1, 2,



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27

8-7b Eigenfilter and FFT Interpolation - Signal 7

PRINT CF MATRIX INTERPOLATION
MULTIPLIER= 0.128E 01 NWIND= 0
NORM(X)= 0.400E 01 NORM(G)= 0.397E 01 NORM(E)= 0.128E 01

I/J	1	2
1	0.782	0.0
2	0.916	-0.134
3	0.782	0.0
4	0.691	0.091
5	0.782	0.0
6	0.863	-0.080
7	0.782	0.0
8	0.701	0.082
9	0.782	0.0
10	0.877	-0.094
11	0.782	0.0
12	0.656	0.126
13	0.782	0.0
14	1.000	-0.218
15	0.782	0.0
16	0.000	0.782
17	-0.782	0.0
18	-1.000	0.218
19	-0.782	0.0
20	-0.656	-0.126
21	-0.782	0.0
22	-0.877	0.094
23	-0.782	0.0
24	-0.701	-0.082
25	-0.782	0.0
26	-0.863	0.080
27	-0.782	0.0
28	-0.691	-0.091
29	-0.782	0.0
30	-0.916	0.134
31	-0.782	0.0
32	-0.368	-0.414

PRINT CF MATRIX INTERPOLATION
MULTIPLIER= 0.117E 01 NWIND= 0
NORM(X)= 0.400E 01 NORM(G)= 0.390E 01 NORM(E)= 0.117E 01

I/J	1	2
1	0.855	0.0
2	0.927	-0.071
3	0.855	0.0
4	0.872	-0.017
5	0.855	0.0
6	0.847	0.008
7	0.855	0.0
8	0.858	-0.003
9	0.855	0.0
10	0.831	0.024
11	0.855	0.0
12	0.877	-0.022
13	0.855	0.0
14	0.939	-0.084
15	0.855	0.0
16	-0.030	0.885
17	-0.855	0.0
18	-1.000	0.145
19	-0.855	0.0
20	-0.888	0.033
21	-0.855	0.0
22	-0.867	0.012
23	-0.855	0.0
24	-0.839	-0.016
25	-0.855	0.0
26	-0.841	-0.015
27	-0.855	0.0
28	-0.853	-0.002
29	-0.855	0.0
30	-0.921	0.065
31	-0.855	0.0
32	-0.436	-0.420

PRINT CF MATRIX INTERPOLATION
MULTIPLIER= 0.128E 01 NWIND=999
NORM(X)= 0.400E 01 NORM(G)= 0.400E 01 NORM(E)= 0.159E 01

I/J	1	2
1	0.780	0.0
2	1.000	-0.220
3	0.780	0.0
4	0.649	0.132
5	0.780	0.0
6	0.883	-0.103
7	0.780	0.0
8	0.685	0.096
9	0.780	0.0
10	0.883	-0.103
11	0.780	0.0
12	0.649	0.132
13	0.780	0.0
14	1.000	-0.220
15	0.780	0.0
16	-0.000	0.780
17	-0.780	0.0
18	-1.000	0.220
19	-0.780	0.0
20	-0.649	-0.132
21	-0.780	0.0
22	-0.883	0.103
23	-0.780	0.0
24	-0.685	-0.096
25	-0.780	0.0
26	-0.883	0.103
27	-0.780	0.0
28	-0.649	-0.132
29	-0.780	0.0
30	-1.000	0.220
31	-0.780	0.0
32	0.000	-0.780

PRINT OF FFT INTERPOLATION
MULTIPLIER= 0.128E 01 NWIND=999
NORM(X)= 0.400E 01 NORM(G)= 0.400E 01 NORM(E)= 0.159E 01

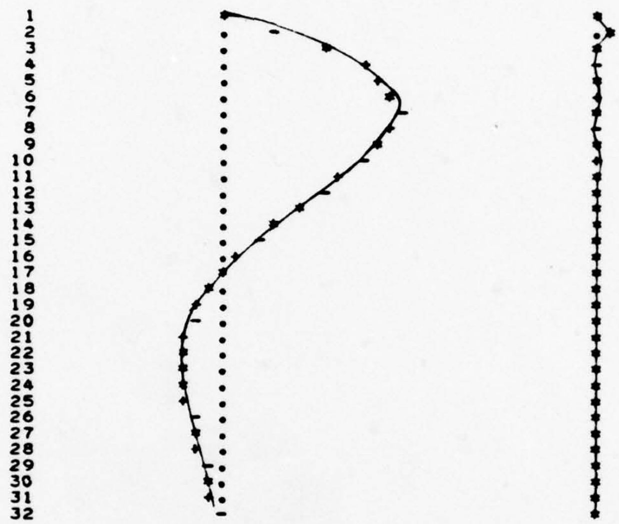
I/J	1	2
1	0.780	0.0
2	1.000	-0.220
3	0.780	0.0
4	0.649	0.132
5	0.780	0.0
6	0.883	-0.103
7	0.780	0.0
8	0.685	0.096
9	0.780	0.0
10	0.883	-0.103
11	0.780	0.0
12	0.649	0.132
13	0.780	0.0
14	1.000	-0.220
15	0.780	0.0
16	-0.000	0.780
17	-0.780	0.0
18	-1.000	0.220
19	-0.780	0.0
20	-0.649	-0.132
21	-0.780	0.0
22	-0.883	0.103
23	-0.780	0.0
24	-0.685	-0.096
25	-0.780	0.0
26	-0.883	0.103
27	-0.780	0.0
28	-0.649	-0.132
29	-0.780	0.0
30	-1.000	0.220
31	-0.780	0.0
32	0.000	-0.780

8-7c Tabular Comparison of Interpolation Schemes - Signal 7

PLOT OF MATRIX INTERPOLATION
MULTIPLIER= 0.526E 00 NWIND= 0
EIGENVECTORS USED IN THE RECONSTRUCTICN
1 2 3 4 5 6 7 8

9 10 11 12 13 14 15 16

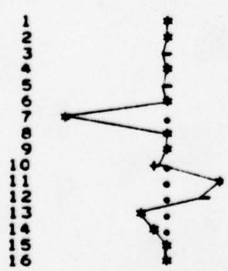
COLUMN 1 IS INTERPOLATED CATA
COLUMN 2 IS THE ERROR
COLUMNS PLOTTED. 1. 2.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27

PLOT OF CORRELATICN COEFFICIENTS
MULTIPLIER= 0.100E 01 NWIND= 0
COLUMNS PLOTTED. 1.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

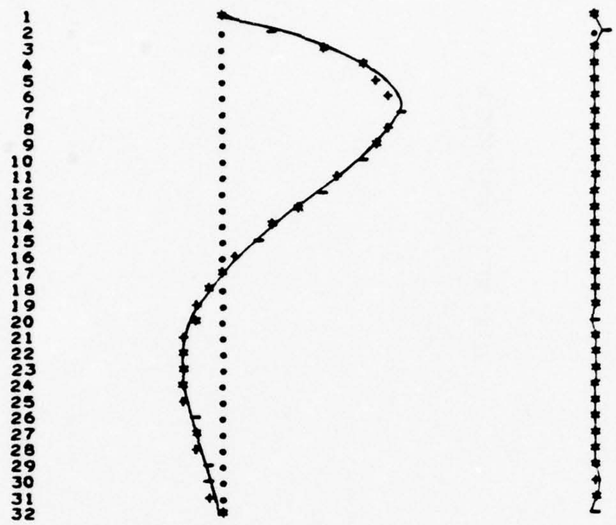
NUMBER OF PEAK-TO-PEAK PRINT CELLS= 19

8-8a Transient Interpolation and Correlation Coefficients - Signal 8

PLOT OF MATRIX INTERPOLATION
MULTIPLIER= 0.526E 00 NWIND= 0
EIGENVECTORS USED IN THE RECONSTRUCTION
0 0 0 0 0 0 7 0

0 0 11 12 13 0 0 0

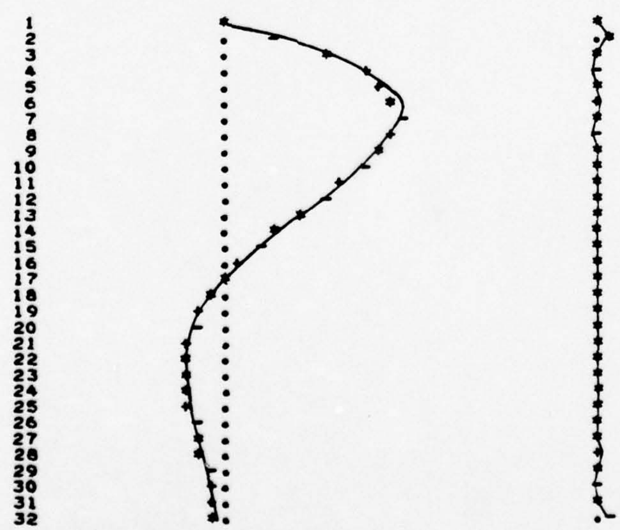
COLUMN 1 IS INTERPOLATED DATA
COLUMN 2 IS THE ERROR
COLUMNS PLOTTED. 1. 2.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27

PLOT OF FFT INTERPOLATION
MULTIPLIER= 0.526E 00 NWIND=999
COLUMN 1 IS INTERPOLATED DATA
COLUMN 2 IS THE ERROR
COLUMNS PLOTTED. 1. 2.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27

8-8b Eigenfilter and FFT Interpolation - Signal 8

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.526E 00 NWIND= 0
NORM(X)= 0.104E 01 NORM(G)= 0.104E 01 NORM(E)= 0.478E-01

I/J	1	2
1	0.0	0.0
2	0.265	0.073
3	0.603	0.0
4	0.833	-0.036
5	0.923	0.0
6	0.965	0.023
7	1.000	0.0
8	0.984	-0.017
9	0.857	0.0
10	0.788	0.013
11	0.687	0.0
12	0.574	-0.011
13	0.436	0.0
14	0.302	0.010
15	0.156	0.0
16	0.059	-0.008
17	0.000	0.0
18	-0.083	0.007
19	-0.134	0.0
20	-0.171	-0.007
21	-0.206	0.0
22	-0.227	0.006
23	-0.223	0.0
24	-0.210	-0.006
25	-0.200	0.0
26	-0.184	0.006
27	-0.153	0.0
28	-0.120	-0.006
29	-0.057	0.0
30	-0.075	0.006
31	-0.044	0.0
32	-0.013	-0.007

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.526E 00 NWIND= 0
NORM(X)= 0.104E 01 NORM(G)= 0.104E 01 NORM(E)= 0.375E-01

I/J	1	2
1	0.0	0.0
2	0.282	0.055
3	0.603	0.0
4	0.805	-0.009
5	0.923	0.0
6	0.988	0.000
7	1.000	0.0
8	0.979	-0.013
9	0.857	0.0
10	0.793	0.008
11	0.687	0.0
12	0.563	-0.000
13	0.436	0.0
14	0.304	0.008
15	0.196	0.0
16	0.054	-0.003
17	0.000	0.0
18	-0.076	0.007
19	-0.134	0.0
20	-0.160	-0.018
21	-0.206	0.0
22	-0.232	0.011
23	-0.223	0.0
24	-0.223	0.007
25	-0.200	0.0
26	-0.179	0.000
27	-0.153	0.0
28	-0.114	-0.012
29	-0.057	0.0
30	-0.087	0.017
31	-0.044	0.0
32	0.007	-0.027

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.526E 00 NWIND=999
NORM(X)= 0.104E 01 NORM(G)= 0.104E 01 NORM(E)= 0.567E-01

I/J	1	2
1	0.0	0.0
2	0.269	0.069
3	0.603	0.0
4	0.830	-0.033
5	0.923	0.0
6	0.968	0.020
7	1.000	0.0
8	0.980	-0.013
9	0.857	0.0
10	0.792	0.009
11	0.687	0.0
12	0.569	-0.006
13	0.436	0.0
14	0.308	0.004
15	0.196	0.0
16	0.092	-0.001
17	0.000	0.0
18	-0.074	-0.001
19	-0.134	0.0
20	-0.181	0.003
21	-0.206	0.0
22	-0.215	-0.006
23	-0.223	0.0
24	-0.224	0.009
25	-0.200	0.0
26	-0.166	-0.013
27	-0.153	0.0
28	-0.144	0.019
29	-0.097	0.0
30	-0.040	-0.030
31	-0.044	0.0
32	-0.080	0.059

PRINT OF FFT INTERPOLATION
MULTIPLIER= 0.526E 00 NWIND=999
NORM(X)= 0.104E 01 NORM(G)= 0.104E 01 NORM(E)= 0.567E-01

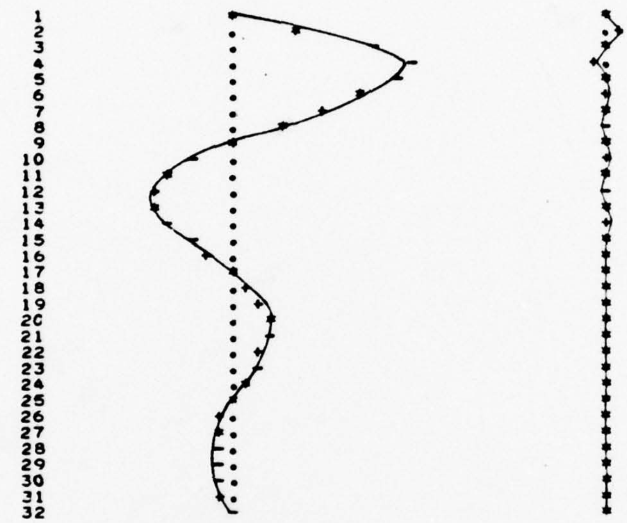
I/J	1	2
1	0.0	0.0
2	0.269	0.069
3	0.603	0.0
4	0.830	-0.033
5	0.923	0.0
6	0.968	0.020
7	1.000	0.0
8	0.980	-0.013
9	0.857	0.0
10	0.792	0.009
11	0.687	0.0
12	0.569	-0.006
13	0.436	0.0
14	0.308	0.004
15	0.196	0.0
16	0.092	-0.001
17	0.000	0.0
18	-0.074	-0.001
19	-0.134	0.0
20	-0.181	0.003
21	-0.206	0.0
22	-0.215	-0.006
23	-0.223	0.0
24	-0.224	0.009
25	-0.200	0.0
26	-0.166	-0.013
27	-0.153	0.0
28	-0.144	0.019
29	-0.097	0.0
30	-0.040	-0.030
31	-0.044	0.0
32	-0.080	0.059

8-8c Tabular Comparison of Interpolation Schemes - Signal 8

PLOT OF MATRIX INTERPOLATION
MULTIPLIER= 0.737E 00 NWIND= 0
EIGENVECTORS USED IN THE RECONSTRUCTION
1 2 3 4 5 6 7 8

9 10 11 12 13 14 15 16

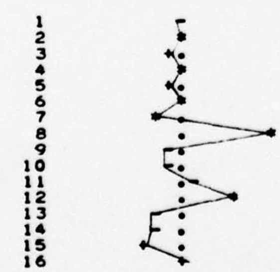
COLUMN 1 IS INTERPOLATED DATA
COLUMN 2 IS THE ERROR
COLUMNS PLOTTED. 1. 2.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27

PLOT OF CORRELATION COEFFICIENTS
MULTIPLIER= 0.100E 01 NWIND= 0
COLUMNS PLOTTED. 1.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

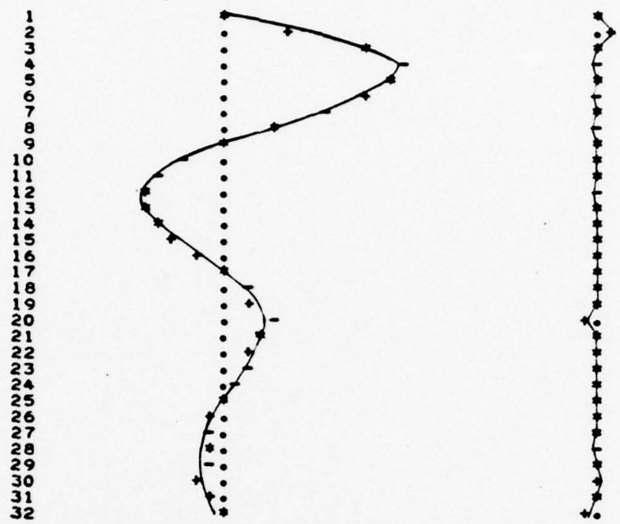
NUMBER OF PEAK-TO-PEAK PRINT CELLS= 19

8-9a Transient Interpolation and
Correlation Coefficients - Signal 9

PLOT OF MATRIX INTERPOLATION
MULTIPLIER= 0.712E 00 NWIND= 0
EIGENVECTORS USED IN THE RECONSTRUCTION
0 0 0 0 0 0 7 8

0 0 0 12 13 14 15 0

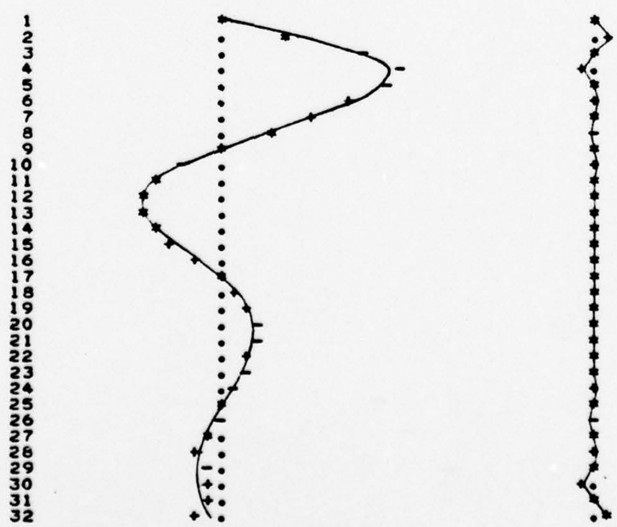
COLUMN 1 IS INTERPOLATED DATA
COLUMN 2 IS THE ERROR
COLUMNS PLOTTED. 1. 2.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27

PLOT OF FFT INTERPOLATION
MULTIPLIER= 0.733E 00 NWIND=999
COLUMN 1 IS INTERPOLATED DATA
COLUMN 2 IS THE ERROR
COLUMNS PLOTTED. 1. 2.



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27

8-9b Eigenfilter and FFT Interpolation - Signal 9

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.737E 00 NWIND= 0
NORM(X)= 0.112E 01 NORM(G)= 0.112E 01 NORM(E)= 0.979E-01

I/J	1	2
1	0.0	0.0
2	0.367	0.106
3	0.756	0.0
4	1.000	-0.053
5	0.933	0.0
6	0.751	0.034
7	0.547	0.0
8	0.255	-0.025
9	0.000	0.0
10	-0.243	0.020
11	-0.376	0.0
12	-0.431	-0.017
13	-0.441	0.0
14	-0.385	0.014
15	-0.256	0.0
16	-0.115	-0.012
17	-0.000	0.0
18	0.095	0.011
19	0.176	0.0
20	0.221	-0.010
21	0.208	0.0
22	0.166	0.009
23	0.122	0.0
24	0.065	-0.009
25	0.000	0.0
26	-0.056	0.008
27	-0.084	0.0
28	-0.092	-0.008
29	-0.058	0.0
30	-0.051	0.009
31	-0.058	0.0
32	-0.018	-0.010

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.712E 00 NWIND= 0
NORM(X)= 0.112E 01 NORM(G)= 0.112E 01 NORM(E)= 0.929E-01

I/J	1	2
1	0.0	0.0
2	0.388	0.102
3	0.824	0.0
4	1.000	-0.020
5	0.566	0.0
6	0.836	-0.024
7	0.566	0.0
8	0.296	-0.017
9	0.000	0.0
10	-0.243	0.012
11	-0.385	0.0
12	-0.438	-0.025
13	-0.456	0.0
14	-0.381	-0.002
15	-0.267	0.0
16	-0.129	-0.003
17	-0.000	0.0
18	0.117	-0.008
19	0.184	0.0
20	0.264	-0.045
21	0.215	0.0
22	0.180	0.002
23	0.126	0.0
24	0.061	0.001
25	0.000	0.0
26	-0.048	-0.003
27	-0.087	0.0
28	-0.084	-0.019
29	-0.102	0.0
30	-0.112	0.026
31	-0.060	0.0
32	0.009	-0.038

PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.733E 00 NWIND=999
NORM(X)= 0.112E 01 NORM(G)= 0.112E 01 NORM(E)= 0.116E 00

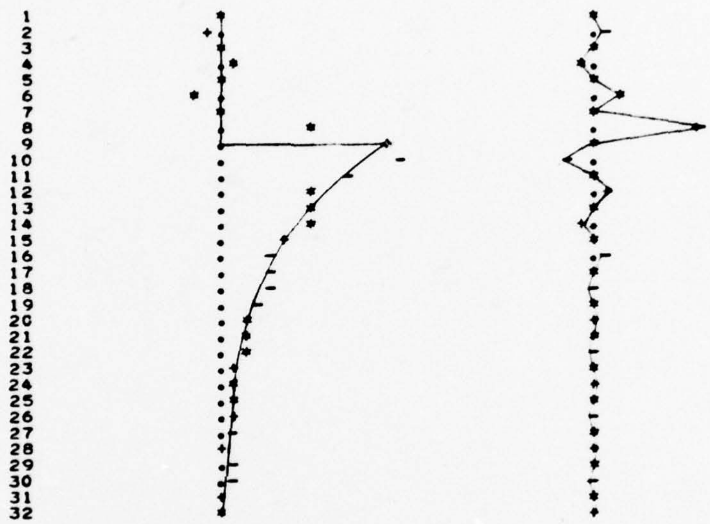
I/J	1	2
1	0.0	0.0
2	0.375	0.101
3	0.759	0.0
4	1.000	-0.049
5	0.937	0.0
6	0.759	0.030
7	0.549	0.0
8	0.291	-0.020
9	0.000	0.0
10	-0.238	0.014
11	-0.378	0.0
12	-0.440	-0.009
13	-0.443	0.0
14	-0.378	0.005
15	-0.260	0.0
16	-0.126	-0.002
17	-0.000	0.0
18	0.107	-0.001
19	0.178	0.0
20	0.208	0.005
21	0.209	0.0
22	0.184	-0.008
23	0.123	0.0
24	0.048	0.013
25	0.000	0.0
26	-0.032	-0.019
27	-0.084	0.0
28	-0.128	0.027
29	-0.099	0.0
30	-0.039	-0.044
31	-0.058	0.0
32	-0.115	0.086

PRINT OF FFT INTERPOLATION
MULTIPLIER= 0.733E 00 NWIND=999
NORM(X)= 0.112E 01 NORM(G)= 0.112E 01 NORM(E)= 0.116E 00

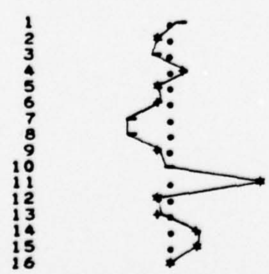
I/J	1	2
1	0.0	0.0
2	0.375	0.101
3	0.759	0.0
4	1.000	-0.049
5	0.937	0.0
6	0.759	0.030
7	0.549	0.0
8	0.291	-0.020
9	0.000	0.0
10	-0.238	0.014
11	-0.378	0.0
12	-0.440	-0.009
13	-0.443	0.0
14	-0.378	0.005
15	-0.260	0.0
16	-0.126	-0.002
17	-0.000	0.0
18	0.107	-0.001
19	0.178	0.0
20	0.208	0.005
21	0.209	0.0
22	0.184	-0.008
23	0.123	0.0
24	0.048	0.013
25	0.000	0.0
26	-0.032	-0.019
27	-0.084	0.0
28	-0.128	0.027
29	-0.099	0.0
30	-0.039	-0.044
31	-0.058	0.0
32	-0.115	0.086

8-9c Tabular Comparison of Interpolation Schemes - Signal 9

```
PLOT OF MATRIX INTERPOLATION
MULTIPLIER= 0.102E 01 NWIND=  0
EIGENVECTORS USED IN THE RECONSTRUCTICN
  1  2  3  4  5  6  7  8

  9 10 11 12 13 14 15 16

COLUMN 1 IS INTERPOLATED CATA
COLUMN 2 IS THE ERFCR
COLUMNS FLOTTED.  1.  2.
```




```
LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27


PLOT OF CORRELATION COEFFICIENTS
MULTIPLIER= 0.100E 01 NWIND=  0
COLUMNS FLOTTED.  1.
```



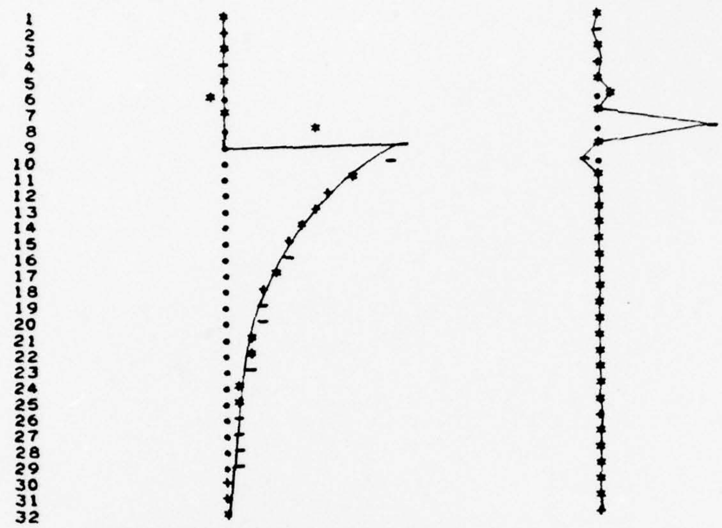

```
LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 19
```

8-10a Transient Interpolation and Correlation Coefficients - Signal 10

PLOT OF MATRIX INTERPOLATION
MULTIPLIER= 0.100E 01 NWIND= 0
EIGENVECTORS USED IN THE RECONSTRUCTION
0 0 0 0 0 0 7 8

0 0 11 0 0 14 15 0

COLUMN 1 IS INTERPOLATED DATA
COLUMN 2 IS THE ERROR
COLUMNS PLOTTED, 1, 2,



LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27

PLOT OF FFT INTERPOLATION
MULTIPLIER= 0.102E 01 NWIND=999
COLUMN 1 IS INTERPOLATED DATA
COLUMN 2 IS THE ERROR
COLUMNS PLOTTED, 1, 2,

LEFT OF CENTER LINE IS NEGATIVE
RIGHT IS POSITIVE

NUMBER OF PEAK-TO-PEAK PRINT CELLS= 27

8-10b Eigenfilter and FFT Interpolation - Signal 10

```
PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.102E 01 NWIND=   0
NORM(X)= 0.147E 01   NORM(G)= 0.146E 01   NORM(E)= 0.703E 00
```

I/J	1	2
1	0.0	0.0
2	-0.057	0.057
3	0.0	0.0
4	0.082	-0.082
5	0.0	0.0
6	-0.147	0.147
7	0.0	0.0
8	0.515	0.629
9	0.978	0.0
10	1.000	-0.163
11	0.716	0.0
12	0.523	0.389
13	0.524	0.0
14	0.508	-0.060
15	0.383	0.0
16	0.282	0.046
17	0.280	0.0
18	0.276	-0.037
19	0.205	0.0
20	0.145	0.031
21	0.150	0.0
22	0.155	-0.027
23	0.110	0.0
24	0.070	0.024
25	0.080	0.0
26	0.090	-0.021
27	0.059	0.0
28	0.030	0.020
29	0.043	0.0
30	0.057	-0.020
31	0.031	0.0
32	-0.000	0.027

```
PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.100E 01 NWIND=   0
NORM(X)= 0.147E 01   NORM(G)= 0.143E 01   NORM(E)= 0.658E 00
```

I/J	1	2
1	0.0	0.0
2	0.015	-0.015
3	0.0	0.0
4	-0.037	0.037
5	0.0	0.0
6	-0.078	0.078
7	0.0	0.0
8	0.525	0.644
9	1.000	0.0
10	0.545	-0.094
11	0.732	0.0
12	0.626	-0.009
13	0.535	0.0
14	0.454	0.003
15	0.392	0.0
16	0.343	-0.009
17	0.267	0.0
18	0.254	-0.009
19	0.210	0.0
20	0.187	-0.008
21	0.153	0.0
22	0.142	-0.010
23	0.112	0.0
24	0.084	0.012
25	0.082	0.0
26	0.057	0.013
27	0.060	0.0
28	0.046	0.005
29	0.044	0.0
30	0.034	0.004
31	0.032	0.0
32	-0.003	0.031

```
PRINT OF MATRIX INTERPOLATION
MULTIPLIER= 0.102E 01 NWIND=999
NORM(X)= 0.147E 01   NORM(G)= 0.146E 01   NORM(E)= 0.696E 00
```

I/J	1	2
1	0.0	0.0
2	-0.054	0.054
3	0.0	0.0
4	0.080	-0.080
5	0.0	0.0
6	-0.146	0.146
7	0.0	0.0
8	0.517	0.630
9	0.981	0.0
10	1.000	-0.161
11	0.718	0.0
12	0.530	0.084
13	0.525	0.0
14	0.502	-0.053
15	0.384	0.0
16	0.293	0.036
17	0.281	0.0
18	0.265	-0.024
19	0.206	0.0
20	0.160	0.015
21	0.150	0.0
22	0.137	-0.008
23	0.110	0.0
24	0.093	0.001
25	0.081	0.0
26	0.064	0.005
27	0.059	0.0
28	0.062	-0.012
29	0.043	0.0
30	0.018	0.019
31	0.032	0.0
32	0.049	-0.022

```
PRINT OF FFT INTERPOLATION
MULTIPLIER= 0.102E 01 NWIND=999
NORM(X)= 0.147E 01   NORM(G)= 0.146E 01   NORM(E)= 0.696E 00
```

I/J	1	2
1	0.0	0.0
2	-0.054	0.054
3	0.0	0.0
4	0.080	-0.080
5	0.0	0.0
6	-0.146	0.146
7	0.0	0.0
8	0.517	0.630
9	0.981	0.0
10	1.000	-0.161
11	0.718	0.0
12	0.530	0.084
13	0.525	0.0
14	0.502	-0.053
15	0.384	0.0
16	0.293	0.036
17	0.281	0.0
18	0.265	-0.024
19	0.206	0.0
20	0.160	0.015
21	0.150	0.0
22	0.137	-0.008
23	0.110	0.0
24	0.093	0.001
25	0.081	0.0
26	0.064	0.005
27	0.059	0.0
28	0.062	-0.012
29	0.043	0.0
30	0.018	0.019
31	0.032	0.0
32	0.049	-0.022

8-10c Tabular Comparison of Interpolation Schemes - Signal 10

Figure	Compile Time	Run Time	||x||	Transient ||g||	Transient ||e||	Selected ||g||	Selected ||e||	Matrix ||g||	Matrix ||e||	FFT ||g||	FFT ||e||
8-1	16.8	11.19	2.83	2.83	.0703	2.84	.104	2.83	.527E-4	2.83	.453E-5
8-2	16.44	11.27	2.83	2.83	.0277	2.83	.0680	2.83	.0595	2.83	.0595
8-3	16.49	10.96	3.33	3.32	.300	3.35	.281	3.33	.0297	3.33	.0297
8-4	16.5	10.95	1.79	1.70	.208	1.69	.150	1.78	.547	1.78	.547
8-5	16.62	11.11	2.30	2.30	.0452	2.32	.0843	2.30	.0571	2.30	.0571
8-6	17.44	11.03	4.00	3.97	1.73	3.93	1.58	4.00	1.59	4.00	1.59
8-7	15.84	10.42	4.00	3.97	1.28	3.90	1.17	4.00	1.59	4.00	1.59
8-8	15.71	10.84	1.04	1.04	.0478	1.04	.0375	1.04	.0567	1.04	.0567
8-9	16.05	10.83	1.12	1.12	.0979	1.12	.0929	1.12	.116	1.12	.116
8-10	16.96	10.69	1.47	1.46	.703	1.43	.658	1.46	.696	1.46	.696

Figure 8-11. Comparison of Interpolation Schemes

Figure	Transient		Selected		Matrix		FFT	
	e_{max}	e_{min}	e_{max}	e_{min}	e_{max}	e_{min}	e_{max}	e_{min}
8-1	.038	.008	.070	.003	0.	0.	0.	0.
8-2	.017	.001	.029	.001	.036	.001	.036	.001
8-3	.264	.019	.210	.002	.018	0.	.018	0.
8-4	.155	.016	.116	.001	.493	0.	.493	0.
8-5	.039	0.	.044	0.	.039	0.	.039	0.
8-6	1.011	.117	.935	.003	.998	.123	.998	.123
8-7	1.001	.102	1.035	.002	.998	.123	.998	.123
8-8	.038	.003	.029	0.	.036	.001	.036	.001
8-9	.078	.006	.073	.001	.074	.001	.074	.001
8-10	.642	.020	.644	0.	.643	.001	.643	.001

Figure 8-12. Comparison of Absolute Errors

Figure	Transient	Selected	Matrix FFT	% $\|\|x\|\|$ Outside Data Interval (Transient)
8-1	.013	.025	0	6.10
8-2	.006	.010	.013	2.82
8-3	.080	.063	.005	37.6
8-4	.091	.069	.277	42.7
8-5	.017	.019	.017	7.98
8-6	.255	.238	.250	119.
8-7	.252	.265	.250	118.
8-8	.037	.028	.035	17.4
8-9	.070	.065	.066	32.9
8-10	.44	.450	.44	207.

Figure 8-13. Comparison of Relative Errors

CHAPTER IX

RECOMMENDATIONS FOR FUTURE WORK

We set out in this work to propose alternatives to linear interpolation - these we have demonstrated in previous chapters through the Whittaker matrix processes. We end our effort by proposing several directions in which matrix interpolating techniques might be extended. As is most often the case with a little knowledge gained, we end by asking more questions than we answered; however, we believe the problem areas are important and have considered each in some detail.

There are four problems that we would like to pursue: (1) extrapolation outside the original data interval using matrix techniques; (2) inverse interpolation for the case when the Whittaker matrix is singular; (3) calculating the derivative of a sample set at each sample point using matrix techniques; (4) recursive or binary interpolation in which the matrix equations are simplified. We formulate and discuss these problems in the following sections.

9.1 Extrapolation Using the Transient Whittaker Matrix

Referring back to Figure 4-1, we note that when $r = 0$, f_4 and f_3 for the two choices of N are really "extrapolated" points. This is inferred in the sense that these points are outside the "original" data. We pose the question of what happens when the interpolants and the extrapolant themselves are interpolated. When N = 4 in Figure 4-1, we would expect this second interpolation to return the original data x_2, x_3 and x_4 plus a new extrapolated point (where x_1 is located in the Figure). Ignoring errors for the moment, we can continue this process using equation 4-8 as follows:

Define

$$f^{(0)} = x \qquad (9\text{-}1)$$

Then,

$$\begin{aligned} f^{(1)} &= Pf^{(0)} \\ &\vdots \\ f^{(i)} &= P^i f^{(0)} \end{aligned} \qquad (9\text{-}2)$$

Where f is a vector and the bracketed superscript (i) infers that the N^{th} element of the vector is the (i^{th}) extrapolant.

Equation 9-2 can be rewritten in terms of the symmetric Whittaker matrix

$$g^{(N-i+1)} = f^{(i)} \qquad (9\text{-}3)$$

or,

$$g^{(i)} = S^i g^{(0)} \qquad (9\text{-}4)$$

Using the orthogonal similarity transformation of Chapter VI

$$g^{(i)} = (Q\Gamma Q^T)^i g^{(0)} \qquad (9\text{-}5)$$

or

$$g^{(i)} = Q\Gamma^i Q^T g^{(0)} \qquad (9\text{-}6)$$

Thus, it seems, extrapolation could be implemented by the programs in the Appendix upon raising the eigenvalues of S to the appropriate power. We now show, however, that this is unwise.

Using the approach in Chapter VII, we try to find the original data vector from our N-times interpolated interpolants; i.e.,

$$x^* = S^{-N} g^{(n)} \tag{9-7}$$

This implies that after N successive interpolations using equation 9-6 we cannot recover x but, rather, some "nearby" vector x*. Then, proceeding as in Section 5 of Chapter VII we find

$$e^{(n)} = S^N(x - x^*) = S^N r \tag{9-8}$$

where $e^{(N)}$ is the error vector for $g^{(N)}$ and r is the residual vector. Then

$$\frac{||e^{(N)}||}{||g^{(N)}||} \leq ||S^N|| \; ||S^{-N}|| \frac{||r||}{||x||} = [k(S)]^N \frac{||r||}{||x||} \tag{9-9}$$

or,

$$\frac{||e^{(N)}||}{||g^{(N)}||} \leq \left|\frac{\gamma_{max}}{\gamma_{min}}\right|^N \frac{||r||}{||x||} \tag{9-10}$$

We note that N = 16, produces a magnification factor on the order of 10^7. Thus it seems that equation 9-6 is unsatisfactory. However, if we expand $g^{(1)}$,

$$g^{(1)} = \begin{pmatrix} g_1 \\ x_N^* \\ x_{N-1}^* \\ \vdots \\ x_2^* \end{pmatrix} \quad (9\text{-}11)$$

and since we know the x_i, we can determine a $\Delta^{(i)}$,

$$\Delta^{(i)} = \begin{pmatrix} \Delta g_1 \\ x_N - x_N^* \\ \vdots \\ x_2 - x_2^* \end{pmatrix} \quad (9\text{-}12)$$

where Δg_1 must be estimated, perhaps from equation 7-31. We can then form a corrected

$$\hat{g}^{(1)} = g^{(1)} - \Delta^{(1)} \quad (9\text{-}13)$$

and continue as

$$g^{(2)} = Q\Gamma^2 Q^T \hat{g}^{(1)} \quad (9\text{-}14)$$

By computing the $g^{(i)}$ at each step in terms of the previous $\hat{g}^{(i-1)}$, it may be possible to significantly reduce the errors in $g^{(N)}$. Needed, of course, are ways to estimate Δg_1 and a rigorous error analysis of the performance of equation 9-6 using corrected data.

9.2 Inverse Interpolation

The inverse problem is simply stated as: given a vector g of interpolants (perhaps computed by some linear approximation scheme), how can we find the original data vector x? If we consider the interpolants as the discrete output from a discrete filter, we ask how do we find the discrete input samples. From our discussions in Chapter VII concerning errors, we know in general that computing $x = S^{-1}g$ returns not x, but rather, x*, some "nearby" vector. Suppose though, that S^{-1} does not exist. This is exactly the problem in the periodic Whittaker matrix with even ordered dimensions. We might ask, then, what this means in terms of our error bounds, especially since g = Sx is a perfectly valid way of finding g given x.

Several important results from Linear Algebra help put this problem into perspective [10, p. 49]. First, Sx = g has NO solution x for certain g. Also, there are non-zero vectors x which satisfy Sx = 0. Finally, there are non-unique solution vectors x to Sx = g.

The first of these results is the most interesting because it implies that we can produce interpolants from some processes which cannot possibly come from interpolating with the Whittaker theory. In fact, [10, p. 48] any vector g which is not a linear combination of the columns of S cannot be produced by Whittaker interpolation. This result obtains from the well known theorem that when the vector g is appended to the columns of S, the rank of the resulting $\hat{S}$ must not change. If g is not a linear combination of the columns of S, then rank $\hat{S}$ will change and thus there is no solution of Sx = g. This all implies that even ordered Whittaker interpolation excludes certain sample sets.

As to the last two results, we know that any vector

$$\bar{x} = \alpha \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \\ \vdots \end{pmatrix} \qquad (9\text{-}15)$$

is a solution of Sx = 0 when order of S is even. This is obvious from the form of equations 5-14 and 5-16 wherein the sum of the column vectors of S is zero if the sign of each column is varied by equation 9-15. Thus, for any solution x of Sx = g, we know $x + \bar{x}$ is also a solution. The vector $\bar{x}$ is simply the sample set from any periodic function whose frequency is exactly equal to the Nyquist rate.

Further study is needed of the error bounds for even ordered interpolation. The mechanics of bounding errors in g = Sx in terms of the actual equation implementing the matrix products is well established in numerical analysis. Also, the numerical errors in computing x (e.g., by Gaussian elimination) are also established. These results can be adapted to the special forms of the Whittaker matrix equations.

9.3 Computing the Derivative

One of many possible numerical formulas for estimating the derivative of a function using its sample set is [28, p. 98]

$$\dot{x}_i \doteq \frac{x_{i+h} - x_{i-h}}{2h} \tag{9-16}$$

where $h = \frac{1}{2}$ the sampling interval if samples from the mid-point Whittaker process are to be used. Our problem here is to outline a study of calculating the derivative at all the sample points using some form of the above equation.

First, recognize that the periodic interpolating equation

$$f = Px \tag{9-17}$$

produces a vector of interpolants at the half steps. Furthermore, if we should multiply f by a permutation matrix of the form

$$T = \begin{bmatrix} 0 & . & . & . & 0 & 1 \\ 1 & 0 & . & . & . & 0 \\ 0 & 1 & 0 & . & . & 0 \\ \vdots & & 1 & . & & \vdots \\ & & & & \ddots & \\ 0 & . & . & 0 & 1 & 0 \end{bmatrix} \tag{9-18}$$

we shift all elements of f to $i - \frac{1}{2}$. Then we can write

$$\dot{x} \doteq \frac{f - Tf}{2h} = \frac{(P - TP)}{2h}x \qquad (9\text{-}19)$$

which is the matrix equation for computing the derivative at all the original data points x_i.

From Chapter V, P is cyclic; then

$$\dot{x} \doteq \frac{(V\Lambda V^* - TV\Lambda V^*)}{2h}x \qquad (9\text{-}20)$$

But T is cyclic too. From Grey [13, pp. 16-21], all cyclic matrices have the same eigenvectors; therefore

$$\dot{x} \doteq \frac{(V\Lambda V^* - V\nabla V^* V\Lambda V^*)}{2h}x \qquad (9\text{-}21)$$

$$= \frac{V[\Lambda(I - \nabla)]V^*}{2h}x \qquad (9\text{-}22)$$

As shown in Chapter V, equation 9-22 is implementable via the FFT algorithm. We also know that the algorithm's numerical problems are expressable as

$$\frac{||\dot{e}||}{||\dot{x}||} \leq \frac{1}{2h} \left|\frac{\lambda_{max}}{\lambda_{min}}\right| \left|\frac{\sigma_{max}}{\sigma_{min}}\right| \frac{||r||}{||x||} \qquad (9\text{-}23)$$

where $\dot{e}$ is the error in the derivatives, the σ_i are eigenvalues of $(I - \nabla)$, and r is $x^* - x$ with x^* computed from the inverse of equation 9-19. We have already shown that $|\lambda_{max}/\lambda_{min}| = 1$ for the periodic case (N = 9), so the stability of the derivative algorithm is

$$\frac{||\dot{e}||}{||\dot{x}||} = \frac{1}{2h} \left|\frac{\sigma_{max}}{\sigma_{min}}\right| \frac{||r||}{||x||} \tag{9-24}$$

We see that small h and/or eigenvalues of T close to 1 ($1 - t_i \rightarrow 0$) cause the error bound to blow up. In fact, for the particular form of T chosen (equation 9-18) we know that the t_i are the N roots of 1. Therefore, $\sigma_{min} = 0$ and equation 9-24 is unbounded.

We propose that other derivative algorithms be investigated to find those with finite (well conditioned) error bounds. Also, the derivative formulae should be extended to the transient Whittaker matrix.

9.4 Recursive (Binary) Interpolation

Again, let the superscript represent the number of data points used in an interpolation scheme. Since S is always square, N also represents the number of interpolants produced. Then, we can form the following sequences

$$g^N = S^N x^N \tag{9-25}$$

$$x^{2N} = T_1^N g^N + T_2^N x^N \tag{9-26}$$

where T_1 and T_2 are permutation matrices which allow g^N and x^N to be added together; for example, when N = 2

$$x^{2\cdot 2} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} g_1^2 \\ g_2^2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1^2 \\ x_2^2 \end{bmatrix} = \begin{bmatrix} x_1^2 \\ g_2^2 \\ x_2^2 \\ g_1^2 \end{bmatrix} \qquad (9\text{-}27)$$

Continuing,

$$g^{2N} = S^{2N}x^{2N} = S^{2N}(T_1^N g^N + T_2^N x^N) = S^{2N}(T_1^N S^N + T_2^N)x^N \qquad (9\text{-}28)$$

$$x^{4N} = T_1^{2N} g^{2N} + T_2^{2N} x^{2N} \qquad (9\text{-}29)$$

$$\begin{aligned} g^{4N} &= S^{4N}x^{4N} = S^{4N}(T_1^{2N}g^{2N}+T_2^{2N}x^{2N}) \\ &= S^{4N}(T_1^{2N}S^{2N} + T_2^{2N})(T_1^N S^N + T_2^N)x^N \end{aligned} \qquad (9\text{-}30)$$

$$\vdots$$

$$g^{mN} = S^{mN}x^{mN} = S^{mN}\left[\prod_{i=1}^{m/2} (T_1^{iN}S^{iN} + T_2^{iN})\right]x^N \qquad (9\text{-}31)$$

where m = 2, 4, 8, 16, ... The dimensions of the resulting matrix equation can be shown as

$$\underset{mN\text{x}1}{\boxed{g}} = \underset{mN \text{ x } mN}{\boxed{S}} \; \underset{mN\text{x}N}{\boxed{\Pi}} \; \underset{N\text{x}1}{\boxed{x}} \qquad (9\text{-}32)$$

All the problems attacked in this dissertation now have analogues in the non-square system expressed by equation 9-32. One potentially fruitful approach to their solution might be to investigate the behavior of the "singular values" [2], [33] of the system using singular value decomposition (SVD). This approach expresses

$$g = U \Sigma V^T x \tag{9-33}$$

where U and V are orthogonal matrices (different dimensions) and Σ is a non-square matrix of the form

$$\Sigma = \begin{bmatrix} \sigma_1 & & 0 \\ & \sigma_2 & \\ 0 & & \ddots \ \sigma_N \\ & 0 & \end{bmatrix}_{mN \times N} \tag{9-34}$$

with the σ_i the square roots of the eigenvalues of $(\Pi^T S^T S \Pi)$. We note that for the nonsingular periodic S, N = 9, $S^T S = I$, and that the σ_i are then the square roots of the eigenvalues of $(\Pi^T \Pi)$. It should be possible to formulate a Singular-Filter, Cross-Correlation Algorithm as an analogue to the Eigen-Filter, Cross-Correlation Algorithm of Chapter VI.

APPENDIX

```
                C   THIS PROGRAM EXERCISES THE SUBROUTINES TO PRODUCE
                C   THE VARIOUS FORMS OF THE WHITTAKER MATRIX WHICH ARE
                C   THEN USED TO INTERPOLATE CATA SETS.
                C   ON INPUT
                C       M IS THE ROW DIMENSION OF THE WHITTTAKER MATRIX.
                C       N IS THE COLUMN DIMENSION OF THE WHITTAKER MATRIX.
                C       NWIND IS THE NUMBER OF WINDOWS USED TO GENERATE MATRIX.
                C           NWIND CAN RANGE FROM 0 TO 999.
                C           NWIND MUST BE ODD, EXCEPT NWIND=0.
                C           N MUST BE A POWER OF 2 IF NWIND=999.
                C       JINT IS A VECTOR OF POINTERS TO EIGENVECTORS TO BE
                C           USED IN INTERPOLATING.
                C       NPLTS IS THE NUMBER OF PLOTS PER OUTPUT PAGE.
                C       JSCALE IS THE NUMBER OF PEAK TO PEAK PRINT CELLS
                C           FOR PLOTTING PLUS ONE TO MINUS ONE.
                C       JSYM IS A VECTOR OF PLOT SYMBOLS.
                C
                C   SET UP STORAGE:
                C
0001                DIMENSION D(64,64),DVEC(64,64),DLMDA(64)
0002                DIMENSION DWORK(64),DWHIT(128),JINT(64)
0003                DIMENSION DLMDAM(1,64),DIT(128,2),DORIG(128)
0004                DIMENSION JCOL(7),JSYM(7)
0005                EQUIVALENCE (DLMDA(1),DLMDAM(1,1))
0006                COMMON TNORMX,TNORMG,TNORME
0007                MDIM=64
0008                NDIM=64
0009                MMDIM=128
0010                NNDIM=2
                C
                C   ON OUTPUT:
                C       NWIND=0 : SAME AS NWIND=1 EXCEPT ALL OUTPUT
                C           RESULTS ARE COMPUTED USING THE MODIFIED
                C           WHITTAKER MATRIX.
                C       NWIND.GE.1.AND.LT.999 : PRINT OF WHITTAKER
                C           MATRIX, PLOT OF ALL EIGENVECTORS, PRINT OF EIGENVALUES
                C           AND EIGENVECTORS, PRINT OF INTERPOLATED DATA AND
                C           ERRORS, PLOT OF CORRELATION COEFFICIENTS, AND PLOT
                C           OF INTERPOLATED DATA AND ERRORS.
                C       NWIND=999 : SAME AS ABOVE PLUS PRINT AND PLOT OF
                C           INTERPOLATED DATA AND ERRORS FROM FFT ROUTINE.
                C
                C   READ AND WRITE INPUT PARAMETERS:
                C
0011                READ(5,101) NPLTS,JSCALE,(JSYM(J),J=1,7)
0012           1000 READ(5,100,END=999) M,N,NWIND,(JINT(J),J=1,N)
0013                WRITE(6,100) MDIM,NDIM,MMDIM,NNDIM
0014                WRITE(6,100) M,N,NWIND,(JINT(J),J=1,N)
0015                WRITE(6,101) NPLTS,JSCALE,(JSYM(J),J=1,7)
                C
                C   GENERATE AND PRINT WHITTAKER MATRIX:
                C
0016                IWIND=NWIND
```

```
      IF(IWIND.EQ.0) IWIND=IWIND+1
      CALL MATRIX(MDIM,NDIM,M,N,D,DMULT,IWIND,DWHIT,MMDIM)
      WRITE(6,104)
      WRITE(6,102)
      WRITE(6,105) DMULT,NWIND
      CALL PRINT(MDIM,NDIM,M,N,D,0)
C
C  MODIFY AND PRINT MODIFIED MATRIX. SKIP IF PARTIAL PERIODIC
C     OR PERIODIC CASE IS INDICATED. (NWIND.GE.1)
C
      IF(NWIND.GE.1) GO TO 10
      D(N,1)=0.
      DC 1 J=2,N
      NWJP2=N-J+2
      NWJP1=N-J+1
      D(NWJP2,J)=0.
      D(NWJP1,J)=0.
    1 CONTINUE
      WRITE(6,104)
      WRITE(6,106)
      WRITE(6,105) DMULT,NWIND
      CALL PRINT(MDIM,NDIM,M,N,D,0)
C
C  GENERATE EIGENVALUES AND EIGENVECTORS. PLOT EIGENVECTORS.
C     NPLTS PER PAGE. AND PRINT EIGENVALUES AND EIGENVECTORS.
C
   10 CALL SYMEIG(NDIM,N,D,DLMDA,DWORK,.TRUE.,DVEC,DMULT,DLMULT,DVMULT)
      NT=N/NPLTS+1
      DC 3 I=1,NT
      DC 2 J=1,NPLTS
      JJ=J+(I-1)*NPLTS
      IF(JJ.GT.N) JJ=0
      JCOL(J)=JJ
    2 CONTINUE
      IF(JCOL(1).EQ.C) GO TO 4
      WRITE(6,104)
      WRITE(6,107)
      WRITE(6,105) DVMULT,NWIND
      CALL PLOT(MDIM,NDIM,M,N,DVEC,JCCL,NPLTS,JSCALE,JSYM)
    3 CONTINUE
    4 WRITE(6,104)
      WRITE(6,109)
      WRITE(6,105) DLMULT,NWIND
      CALL PRINT(1,NDIM,1,N,DLMDAM,0)
      WRITE(6,11C)
      WRITE(6,105) DVMULT,NWIND
      CALL PRINT(MDIM,NDIM,M,N,DVEC,0)
C
C  GENERATE DATA SET:
C
      M2=MAX0(M,N)*2
      CALL SIGNAL(MMDIM,M2,DORIG)
C
```

```
          C  INTERPOLATE USING THE DECCMPOSED WHIT MATRIX. PRINT
          C     INTERPOLATED DATA AND ERRORS; PLOT CORRELATION
          C     CCEFFICIENTS; PLOT INTERPOLATED DATA AND ERRORS.
          C
0058            NTRL=1
0059            IF(NWIND.GE.1) NTRL=0
0060            CALL INTERP(NDIM,M2,N,MMDIM,NNDIM,DVEC,CLMDA,DORIG,
               1DIT,DWORK,JINT,NTRL,DIMULT,DMULT,DLMULT,DVMULT)
0061            WRITE(6,104)
0062            WRITE(6,117)
0063            WRITE(6,105) DIMULT,NWIND
0064            WRITE(6,120) TNORMX,TNCRMG,TNORME
0065            CALL PRINT(MMDIM,NNDIM,M2,2,DIT,0)
0066            WRITE(6,104)
0067            WRITE(6,111)
0068            CCN=1.
0069            WRITE(6,105) CON,NWIND
0070            JCOL(1)=1
0071            JCOL(2)=2
0072            CALL PLOT(MDIM,1,M,1,DWORK,JCOL,1,19,JSYM)
0073            WRITE(6,104)
0074            WRITE(6,114)
0075            WRITE(6,105) DIMULT,NWIND
0076            WRITE(6,115)
0077            WRITE(6,116) (JINT(J),J=1,N)
0078            WRITE(6,112)
0079            WRITE(6,113)
0080            CALL PLCT(MMDIM,NNDIM,M2,2,DIT,JCOL,2,0,JSYM)
          C
          C  INTERPOLATE USING FAST FOURIER TRANSFORM TECHNIQUE.
          C     SKIP IF LESS THAN EXACT PERIODICITY IS INDICATED.
          C     (NWIND.LT.959). PRINT INTERPOLATED DATA AND ERRORS.
          C     PLOT INTERPOLATED CATA AND ERRORS.
          C
0081            IF(NWIND.LT.999) GO TO 99
0082            CALL FFTINT(NDIM,N,MMDIM,NNDIM,DIT,DWHIT,DLMDA,
               1DCRIG,DWORK,DMULT,DFMULT)
0083            WRITE(6,104)
0084            WRITE(6,121)
0085            WRITE(6,105) DFMULT,NWIND
0086            WRITE(6,120) TNORMX,TNORMG,TNORME
0087            CALL PRINT(MMDIM,NNDIM,M2,2,DIT,0)
0088            WRITE(6,104)
0089            WRITE(6,119)
0090            WRITE(6,105) DFMULT,NWIND
0091            WRITE(6,112)
0092            WRITE(6,113)
0093            CALL PLCT(MMDIM,NNDIM,M2,2,DIT,JCOL,2,0,JSYM)
          C
          C
          C
0094         99 WRITE(6,104)
0095            GC TO 1000
```

```
      100 FORMAT(19I3)
      101 FORMAT(2I3,7A1)
      102 FORMAT(15X,'PRINT OF WHITTAKER MATRIX')
      104 FORMAT(1H1,//////)
      105 FORMAT(15X,'MULTIPLIER=',E10.3,' NWIND=',I3)
      106 FORMAT(15X,'PRINT OF MODIFIED MATRIX.')
      107 FORMAT(15X,'PLOT OF EIGENVECTORS.')
      109 FORMAT(15X,'PRINT OF EIGENVALUES.')
      110 FORMAT(//,15X,'PRINT OF EIGENVECTORS.')
      111 FORMAT(15X,'PLOT OF CORRELATION COEFFICIENTS')
      112 FORMAT(15X,'COLUMN 1 IS INTERPOLATED DATA')
      113 FORMAT(15X,'COLUMN 2 IS THE ERROR')
      114 FORMAT(15X,'PLOT OF MATRIX INTERPOLATION')
      115 FORMAT(15X,'EIGENVECTORS USED IN THE RECONSTRUCTION')
      116 FORMAT(15X,8I3,/)
      117 FORMAT(15X,'PRINT OF MATRIX INTERPOLATION')
      119 FORMAT(15X,'PLOT OF FFT INTERPOLATION')
      120 FORMAT(15X,'NORM(X)=',E10.3,'  NORM(G)=',E10.3,'  NORM(E)=',E10.3)
      121 FORMAT(15X,'PRINT OF FFT INTERPOLATION')
      999 STOP
          END
```

```
FORTRAN IV G LEVEL  21                FFT                DATE = 77062           06/12/54            PAGE 0001

0001                SUBROUTINE FFT(MMDIM,NDIM,N,FR,FI)
              C     FAST FOURIER TRANSFORM USING TIME DECOMPOSITION
              C     WITH INPUT BIT REVERSAL
              C     DATA IS IN FR (REAL) AND FI (IMAGINARY) ARRAYS
              C     COMPUTATION IS IN PLACE, OUTPUT REPLACES INPUT
              C     NUMBER OF POINTS MUST BE A POWER OF 2.
0002                DIMENSION FR(MMDIM),FI(NDIM)
0003                MR=0
0004                NN=N-1
0005                DO 2 M=1,NN
0006                L=N
0007           1    L=L/2
0008                IF(MR+L.GT.NN) GO TO 1
0009                MR=MOD(MR,L)+L
0010                IF(MR.LE.M) GO TO 2
0011                TR=FR(M+1)
0012                FR(M+1)=FR(MR+1)
0013                FR(MR+1)=TR
0014                TI=FI(M+1)
0015                FI(M+1)=FI(MR+1)
0016                FI(MR+1)=TI
0017           2    CONTINUE
0018                L=1
0019           3    IF(L.GE.N) RETURN
0020                ISTEP=2*L
0021                EL=L
0022                DO 4 M=1,L
0023                A=3.141593*FLOAT(1-M)/EL
0024                WR=COS(A)
0025                WI=SIN(A)
0026                DO 4 I=M,N,ISTEP
0027                J=I+L
0028                TR=WR*FR(J)-WI*FI(J)
0029                TI=WR*FI(J)+WI*FR(J)
0030                FR(J)=FR(I)-TR
0031                FI(J)=FI(I)-TI
0032                FR(I)=FR(I)+TR
0033           4    FI(I)=FI(I)+TI
0034                L=ISTEP
0035                GO TO 3
0036                END
```

```
0001          SUBROUTINE FFTINT(NDIM,N,MMDIM,NNDIM,DIT,DLR,DLI,
             1DCR,DOI,DMULT,DFMULT)
          C   INTERPOLATES DATA VECTOR DORIG, BY TAKING THE FFT OF THE FIRST
          C   ROW OF WHITTAKER MATRIX, MULTIPLYING BY FFT OF DORIG, AND
          C   TAKING THE INVERSE FFT OF THE PRODUCT. RESULTS ARE
          C   PERMUTED WHICH THE PROGRAM THEN CORRECTS.
          C   ON INPUT:
          C      NDIM IS THE DIMENSION OF DLI AND DLR IN MAIN.
          C      N IS THE NUMBER OF DATA POINTS TO BE INTERPOLATED.
          C      MMDIM IS THE DIMENSION OF DLR AND DOR AND THE ROW
          C         DIMENSION OF DIT IN MAIN.
          C      NNDIM IS THE COLUMN DIMENSION OF DIT IN MAIN.
          C      DLR IS THE FIRST ROW OF WHITTAKER MATRIX.
          C      DLI IS WORK VECTOR.
          C      DCR IS THE VECTOR OF ORIGINAL DATA.
          C      DCI IS WORK VECTOR.
          C   ON OUTPUT:
          C      DIT IS AN N*2 BY 2 MATRIX OF INTERPOLATED DATA
          C         AND ERRORS. COLUMN 1 IS THE INTERPOLATED DATA.
          C         COLUMN 2 IS THE VECTOR OF ERRORS.
          C      DMULT IS SCALE FACTOR FOR DLR.
          C      DFMULT IS SCALE FACTOR FOR INTERPOLATED DATA.
0002          DIMENSION DIT(MMDIM,NNDIM),DLR(MMDIM),DLI(NDIM)
0003          DIMENSION DCR(MMDIM),DCI(NDIM)
0004          COMMON TNORMX,TNORMG,TNORME
          C
          C   GENERATE FFT OF DATA AND FIRST ROW OF WHITTAKER MATRIX.
          C
0005          DO 1 I=1,N
0006          IE=I*2
0007          IO=IE-1
0008          DIT(IO,1)=DCR(IO)
0009          DIT(IO,2)=0.
0010          DIT(IE,1)=DOR(IE)
0011          DLI(I)=0.
0012          DCI(I)=0.
0013        1 CONTINUE
0014          DO 2 I=1,N
0015          IO=2*I-1
0016          DCR(I)=DIT(IO,1)
0017        2 CONTINUE
0018          CALL FFT(MMDIM,NDIM,N,DLR,DLI)
0019          CALL FFT(MMDIM,NDIM,N,DOR,DOI)
          C
          C   GENERATE PRODUCTS.
          C
0020          DO 3 I=1,N
0021          TR=DLR(I)
0022          TI=DLI(I)
0023          DLR(I)=TR*DOR(I)+TI*DOI(I)
0024          DLI(I)=TI*DCR(I)-TR*DOI(I)
0025        3 CONTINUE
          C
```

```
C     GENERATE INVERSE FFT. REMOVE AFFECTS OF PERMUTATION
C         AND CALIBRATE.
C
          CALL FFT(MMDIM,NDIM,N,DLR,DLI)
          CAL=DMULT/FLOAT(N)
          N2=N*2
          DCR(N2)=DLR(1)*CAL
          DC 4 I=2,N
          IEM2=2*I-2
          DCR(IEM2)=DLR(I)*CAL
        4 CCNTINUE
C
C     INTERLEAVE INTERPOLANTS AND ORIGINAL CATA. CCMPUTE ERRORS.
C
          TNORMX=0.
          TNCRMG=0.
          TNORME=0.
          DC 5 I=1,N
          IE=2*I
          IC=IE-1
          CIT(IE,2)=DIT(IE,1)-DOR(IE)
          DIT(IE,1)=DOR(IE)
          TNORMX=TNORMX+DIT(IO,1)**2
          TNORMG=TNORMG+DIT(IE,1)**2
          TNORME=TNORME+DIT(IE,2)**2
        5 CCNTINUE
          TNORMX=SQRT(TNORMX)
          TNORMG=SQRT(TNORMG)
          TNORME=SQRT(TNORME)
C
C     NORMALIZE THE RESULTS:
C
          DFMULT=C.
          DC 11 I=1,N2
          DC 10 J=1,2
          DF=ABS(DIT(I,J))
          IF(DFMULT.LT.DF) DFMULT=DF
       10 CCNTINUE
       11 CCNTINUE
          DC 21 J=1,2
          DC 20 I=1,N2
          IF(DFMULT.EC.0) GO TO 21
          DIT(I,J)=DIT(I,J)/DFMULT
       20 CONTINUE
       21 CCNTINUE
          RETURN
          END
```

```
0001             SUBROUTINE INTERP(NDIM,M,N,MMDIM,NNDIM,DVEC,DLMDA,DORIG,
                1 DIT,DSGMA,JCOL,NTRL,DIMULT,DMULT,DLMULT,DVMULT)
           C   INTERPOLATES THE ORIGINAL DATA VECTOR, DORIG. COMPUTES
           C   Y=VLUX WHERE Y IS THE VECTOR OF INTERPOLANTS, V IS DVEC,
           C   L IS DLMDA,U IS V TRANSPOSED,AND X IS THE ORIGINAL DATA.
           C   ON INPUT:
           C       NDIM IS COLUMN DIMENSION OF DVEC IN MAIN PROGRAM.
           C       M IS THE NUMBER OF DATA POINTS IN DORIG.
           C       N IS NUMBER OF COLUMNS IN DVEC.
           C       MMDIM IS THE ROW DIMENSION OF DIT IN MAIN PROGRAM.
           C       NNDIM IS THE COLUMN DIMENSION OF DIT IN MAIN PROGRAM.
           C       DVEC IS MATRIX OF EIGENVECTORS FROM SYMEIG.
           C       DLMDA IS THE VECTOR OF EIGENVALUES FROM SYMEIG.
           C       DORIG IS AN M VECTOR OF ORIGINAL DATA. THE ODD POINTS
           C          ARE THE DATA TO BE INTERPOLATED. THE EVEN POINTS
           C          ARE THE EXACT VALUES AGAINST WHICH THE INTERPOLATED
           C          POINTS ARE TO BE COMPARED.
           C       JCOL IS THE N VECTOR OF INDEXES INDICATING WHICH EIGEN-
           C          VECTORS AND EIGENVALUES ARE TO BE USED IN Y=VLUX.
           C       NTRL.NE.0 MEANS DVEC IS FROM A MODIFIED WHITTAKER
           C          MATRIX. INTERP REMOVES THE AFFECTS OF THE MOD.
           C       DMULT IS THE SCALE FACTOR OF THE ORIGINAL WHITTAKER
           C          MATRIX.
           C       DLMULT IS THE EIGENVALUE SCALE FACTOR.
           C       DVMULT IS THE EIGENVECTOR SCALE FACTOR.
0002             DIMENSION DVEC(NDIM,NDIM),DLMDA(NDIM),DORIG(MMDIM)
0003             DIMENSION DSGMA(NDIM),JCOL(NDIM),DIT(MMDIM,NNDIM)
0004             COMMON TNORMX,TNORMG,TNORME
           C   ON OUTPUT:
           C       DIT IS AN M BY 2 MATRIX OF INTERPOLATED DATA AND
           C          DIFFERENCES. COLUMN 1 IS THE VECTOR OF INTERPOLATED
           C          POINTS. COLUMN 2 IS THE DIFFERENCE BETWEEN THE
           C          INTERPOLATED POINTS AND THE ORIGINAL DATA.
           C       DSGMA IS THE N VECTOR OF CORRELATION COEFFICIENTS
           C          FORMED BY MULTIPLYING THE ROWS OF DVEC TRANSPOSED
           C          BY THE ODD POINTS IN DORIG. DSGMA=UX AS DESCRIBED
           C          ABOVE.
           C       DIMULT IS THE SCALE FACTOR FOR THE INTERPOLATED DATA.
           C
           C   INTERPOLATE
           C
0005             CON=DLMULT*(DVMULT**2)
0006             DO 30 I=1,N
0007             DSGMA(I)=0.
0008             IF(JCOL(I).EQ.0) GO TO 30
0009             DO 10 J=1,N
0010             JODD=2*J-1
0011             DSGMA(I)=DVEC(J,I)*DORIG(JODD)+DSGMA(I)
0012          10 CONTINUE
0013             DLMDA(I)=DSGMA(I)*DLMDA(I)*CON
0014          30 CONTINUE
0015             DO 41 I=1,N
0016             IEVEN=2*I
```

```
      IODD=IEVEN-1
      IIEVEN=2*N-IEVEN+2
      DIT(IIEVEN,1)=0.
      DIT(IODD,1)=DORIG(IODD)
      DO 40 J=1,N
      IF(JCOL(J).EQ.0) GO TO 40
      DIT(IIEVEN,1)=DVEC(I,J)*DLMDA(J)+DIT(IIEVEN,1)
   40 CONTINUE
   41 CONTINUE
C
C  REMOVE AFFECTS OF MODIFIED WHITTAKER MATRIX. INTERLEAVE
C      RESULTS WITH ORIGINAL DATA AND COMPUTE ERRORS.
C
      IF(NTRL.EQ.0) GO TO 55
      NM1=N-1
      DO 50 I=1,NM1
      IEVEN=2*I
      IODD=IEVEN-1
      IEP1=IEVEN+1
      DIT(IEVEN,1)=DIT(IEVEN,1)+DORIG(IODD)+DORIG(IEP1)
   50 CONTINUE
      N2=2*N
      N2M1=N2-1
      DIT(N2,1)=DIT(N2,1)+DORIG(N2M1)
   55 TNORMX=0.
      TNORMG=0.
      TNORME=0.
      DO 60 I=1,N
      IEVEN=2*I
      IODD=IEVEN-1
      DIT(IEVEN,1)=DIT(IEVEN,1)*DMULT
      DIT(IEVEN,2)=DORIG(IEVEN)-DIT(IEVEN,1)
      DIT(IODD,2)=0.
      TNORMX=TNORMX+DIT(IODD,1)**2
      TNORMG=TNORMG+DIT(IEVEN,1)**2
      TNORME=TNORME+DIT(IEVEN,2)**2
   60 CONTINUE
      TNORMX=SQRT(TNORMX)
      TNORMG=SQRT(TNORMG)
      TNORME=SQRT(TNORME)
C
C  NORMALIZE THE RESULTS.
C
      N2=2*N
      DIMULT=0.
      DS=0.
      DO 70 I=1,N
      IO=2*I-1
      DS=DS+DORIG(IO)*DORIG(IO)
   70 CONTINUE
      DSMULT=SQRT(DS)
      DO 72 I=1,N2
      DO 71 J=1,2
```

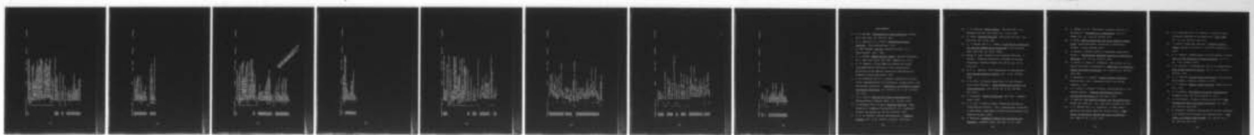
```
      DI=ABS(DIT(I,J))
      IF(DIMULT.LT.DI) DIMULT=DI
   71 CCNTINUE
   72 CCNTINUE
      DC 80 I=1,N
      DSGMA(I)=DVMULT*DSGMA(I)/DSMULT
   80 CCNTINUE
   81 DO 83 J=1,2
      DO 82 I=1,N2
      IF(DIMULT.EQ.0) GO TO 83
      DIT(I,J)=DIT(I,J)/DIMULT
   82 CCNTINUE
   83 CCNTINUE
      RETURN
      END
```

```
0001          SLBROUTINE MATRIX(MDIM,NDIM,M,N,D,DMULT,NWIND,DW,MMDIM)
        C
        C  GENERATES WHITTAKER MATRIX. USES EQUATION (4-29) FOR
        C  TRANSIENT CASE. AND EQUATIONS (5-11) OR (5-12) FOR
        C  PERIODIC CASE. MATRIX NEED NOT BE SQUARE.
        C  ON INPUT:
        C     MDIM IS ROW DIMENSION OF D IN MAIN PROGRAM.
        C     NDIM IS CJLUMN DIMENSION OF D IN MAIN PROGRAM.
        C     M IS NUMBER OF ROWS IN D.
        C     N IS NUMBER OF COLUMNS IN D.
        C     NWIND (NWIND.GE.999) PERIODIC CASE. (NWIND.LT.999)
        C        TRANSIENT CASE. NWIND MUST BE ODD.
        C     DW IS A WORK VECTOR.
        C     MMDIM IS DIMENSION OF DW IN MAIN PROGRAM.
        C  ON CLTPUT:
        C     MATRIX D SCALED TO PLUS OR MINUS ONE.
        C     DMULT IS SCALE FACTOR.
        C     DW CONTAINS THE FIRST M2 POINTS OF THE WHITTAKER
        C        SEQUENCE.
        C
0002          CIMENSION D(MDIM,NDIM),DW(MMDIM)
0003          CATA DPI/3.141593/
0004          M2=MAX0(M,N)*2
0005          M2D2=M2/2
0006          IF(NWIND.LT.999) GO TO 4
        C
        C  DETERMINE IF ORDER OF D IS EVEN OR ODD.
        C
0007          JCRDER=MOD(N,2)
        C
        C  GENERATE FIRST M2D2 POINTS OF PERIODIC SEQUENCE.
        C
0008          ISIGN=-1
0009          DN=FLOAT(N)
0010          D2N=FLOAT(2*N)
0011          DC 3 I=1,M2D2
0012          DSIGN=FLOAT(ISIGN)
0013          DARG=DPI*FLCAT(1-2*I)/D2N
0014          IF(JORDER.EQ.1) GO TO 1
0015          DW(I)=DSIGN*COTAN(DARG)/DN
0016          GC TO 2
0017        1 DW(I)=DSIGN/(SIN(DARG)*DN)
0018        2 ISIGN=-1*ISIGN
0019        3 CCNTINUE
0020          GC TO 7
        C
        C  GENERATES FIRST M2D2 POINTS OF TRANSIENT SEQUENCE.
        C
0021        4 NNWIND=(NWIND+3)/2
0022          DC 6 I=1,M2D2
0023          DW(I)=0.
0024          CC 5 JR=1,NWIND
0025          IEXP=N*(NNWIND-JR)-I
```

```
          DW(I)=(-1.)**IEXP/(2.*FLOAT(IEXP)+1.)+DW(I)
        5 CCNTINUE
          DW(I)=2.*DW(I)/DPI
        6 CCNTINUE
C
C     GENERATE REST OF MATRIX.
C
        7 DMULT=ABS(DW(N))
          DC 8 I=1,M2D2
          DW(I)=DW(I)/DMULT
        8 CCNTINUE
          M2D2=M2/2
          DC 81 I=1,M2D2
          M2MIP1=M2-I+1
          DW(M2MIP1)=DW(I)
       81 CCNTINUE
          DC 11 J=1,N
          DC 10 I=1,M
          II=I+J-1
          D(I,J)=DW(II)
       10 CONTINUE
       11 CCNTINUE
          RETURN
          END
```

AD-A061 030 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO F/G 9/3
DIGITAL SIGNAL INTERPOLATION USING MATRIX TECHNIQUES AND THE WH--ETC(U)
MAY 78 J C WHEELER
UNCLASSIFIED AFIT-CI-78-120D NL

3 OF 3
AD
A061030



END
DATE
FILMED
1 -79
DDC

```
FORTRAN IV G LEVEL  21              PLOT              DATE = 77062         06/12/54          PAGE 0001

 0001            SUBROUTINE PLOT(MDIM,NDIM,M,N,D,JCOL,LCNT,JSIZE,JSYM)
           C  PLOTS UP TO 7 COLUMNS OF MATRIX D. MARGINS ARE SET FOR
           C  USE WITH 8.5X11 INCH PAPER.
           C  ON INPUT:
           C      MDIM IS ROW DIMENSION OF D IN MAIN PROGRAM.
           C      NDIM IS COLUMN DIMENSION OF D IN MAIN PROGRAM.
           C      M IS NUMBER OF ROWS IN D.
           C      N IS NUMBER OF COLUMNS IN D.
           C      D IS MATRIX SCALED TO PLUS OR MINUS ONE.
           C      JCOL(J),J=1,7 SPECIFIES COLUMNS OF D TO BE PLOTTED.
           C         JCOL(J) MAY HAVE ANY VALUE GE.0.AND.LE.N.
           C      LCNT SPECIFIES TOTAL NUMBER OF PLOTS. MAX=7.
           C      JSIZE IS THE NUMBER OF PRINT CELLS FOR PLOTTING PLUS
           C         ONE TO MINUS ONE. JSIZE.EQ.0 CAUSES PLOT TO ADJUST
           C         TO OPTIMUM. JSIZE.NE.0 CAUSES PLOT TO USE INPUT
           C         VALUE.
           C      JSYM(J),J=1,7 IS A VECTOR OF PLOT SYMBOLS.
           C         J.EQ.1.OR.2.OR.3 SPECIFIES THAT PLOTTED POINT
           C         IS TO THE LEFT, CENTER, OR RIGHT OF THE PRINT CELL.
           C         J.EQ.4 IS THE BACKGROUND (NORMALLY BLANK).
           C         J.EQ.5 IS THE CENTER LINE FOR EACH PLOT.
           C         J.EQ.6 MEANS PLOTTED POINT IS QUESTICNABLE; IE.
           C         EXCEEDS LIMITS OF PLOT ROUTINE AND IS SET TO LIMIT.
           C  ON OUTPUT:
           C      PLOTS OF UP TO 7 COLUMNS OF D.
           C      PRINTS N+4 LINES PER CALL.
           C
 0002            DIMENSION D(MDIM,NDIM),JCOL(7),JSYM(7),JPLT(90),JSCL(7)
 0003            DATA JSCL/55,27,17,19,15,13,11/
 0004            WRITE(6,100)(JCOL(J),J=1,LCNT)
 0005            WRITE(6,104)
           C
           C  COMPUTE NUMBER OF PRINT CELLS PER PLOT.
           C
 0006            JSCALE=JSIZE
 0007            IF(JSIZE.EQ.0) JSCALE=JSCL(LCNT)
           C
           C  GENERATE PRINT VECTOR.
           C
 0008            DO 4 I=1,M
 0009            DO 2 J=1,90
 0010            JPLT(J)=JSYM(4)
 0011          2 CONTINUE
 0012            DO 3 L=1,LCNT
 0013            JJ=JCOL(L)
 0014            IF(JJ.EQ.0) GO TO 3
 0015            DARG=(1.+D(I,JJ))*FLOAT(JSCALE)/2.
 0016            JB=(JSCALE+2)*(L-1)+1
 0017            JM=JB+JSCALE/2
 0018            J=JB+INT(DARG)
 0019            JPLT(JM)=JSYM(5)
 0020            IF(J.LT.1) JPLT(1)=JSYM(6)
 0021            IF(J.GT.90) JPLT(90)=JSYM(6)
```

```
            IF(J.LT.1.OR.J.GT.90) GO TO 3
            JPLT(J)=JSYM(2)
            DARG=DARG-FLOAT(INT(DARG))
            IF(DARG.GT..67) JPLT(J)=JSYM(3)
            IF(DARG.LE..33) JPLT(J)=JSYM(1)
          3 CONTINUE
            WRITE(6,101) I,(JPLT(J),J=1,90)
        101 FORMAT(15X,I4,1X,90A1)
          4 CONTINUE
            WRITE(6,102)
            WRITE(6,103)
            WRITE(6,108) JSCALE
            RETURN
      C
      C
      C
        100 FORMAT(15X,'COLUMNS PLOTTED',7(',',I3))
        102 FORMAT(/,15X,'LEFT OF CENTER LINE IS NEGATIVE')
        103 FORMAT(15X,'RIGHT IS POSITIVE')
        104 FORMAT(/)
        108 FORMAT(/,15X,'NUMBER OF PEAK-TO-PEAK PRINT CELLS=',I3)
            END
```

```
0001          SUBROUTINE PRINT(MDIM,NDIM,M,N,D,IROW)
        C
        C  PRINTS MATRIX D OR SELECTED ROWS OF D.
        C  MARGINS ARE SET FOR USE WITH 8.5X11 INCH PAPER.
        C  ON INPUT:
        C     MDIM IS THE ROW DIMENSION OF D IN MAIN PROGRAM.
        C     NDIM IS THE COLUMN DIMENSION OF D IN MAIN PROGRAM.
        C     M IS THE NUMBER OF ROWS OF D.
        C     N IS THE NUMBER OF COLUMNS OF D.
        C     D IS THE MATRIX SCALED TO PLUS OR MINUS ONE.
        C     (IROW.NE.0) OVERRIDES MATRIX PRINT WITH ROW PRINT.
        C  ON OUTPUT:
        C     (N.LE.16.AND.IROW.EQ.0) PRINT MATRIX D EACH CALL.
        C         PRINTS M+2 LINES
        C     (OTHERWISE) PRINT IROW (ROW 1 (IROW.EQ.0)) OF D EACH
        C        CALL. PRINTS 3*N/8 LINES WHEN N IS A MULTIPLE OF
        C        8. PRINTS 3*(N/8+1) LINES OTHERWISE.
0002          DIMENSION D(MDIM,NDIM)
0003          IF(N.LE.16.AND.IROW.EQ.0) GO TO 2
        C
        C  ROW PRINT
        C
0004          ITEMP=IROW
0005          IF(IROW.EQ.0) ITEMP=1
0006          LCOL=1+N/8
0007          DO 1 L=1,LCOL
0008          JB=1+(L-1)*8
0009          JE=L*8
0010          JE=MINO(JE,N)
0011          IF(JE.LT.JB) RETURN
0012          WRITE(6,100) (J,J=JB,JE)
0013      100 FORMAT(/,16X,'I/J',1X,8(I3,4X))
0014          WRITE(6,110) ITEMP,(D(ITEMP,J),J=JB,JE)
0015      110 FORMAT(14X,I3,1X,8(F6.3,1X))
0016        1 CONTINUE
0017          RETURN
        C
        C  MATRIX PRINT.
        C
0018        2 IF(N.GT.8) GO TO 4
0019          WRITE(6,100) (J,J=1,N)
0020          DO 3 I=1,M
0021          WRITE(6,110) I,(D(I,J),J=1,N)
0022        3 CONTINUE
0023          RETURN
0024        4 WRITE(6,120) (J,J=1,N)
0025      120 FORMAT(11X,'I/J',1X,16(I3,4X))
0026          DO 5 I=1,M
0027          WRITE(6,130) I,(D(I,J),J=1,N)
0028      130 FORMAT(9X,I3,1X,16(F6.3,1X))
0029        5 CONTINUE
0030          RETURN
0031          END
```

```
FORTRAN IV G LEVEL  21                SIGNAL            DATE = 77062          06/12/54                PAGE 0001
0001              SUBROUTINE SIGNAL(MMDIM,M,DORIG)
            C  GENERATES M POINTS OF SOME FUNCTION WHICH HAS MAX AMPLITUDE
            C  OF PLUS OR MINUS ONE. AND WHICH HAS DURATION OF M POINTS.
0002              DIMENSION DORIG(MMDIM)
0003              TWOPI=6.283185
0004              DO 10 I=1,M
0005              MD4=M/4
0006              M3D4=M*3/4
0007              IF(I.LE.MD4) DORIG(I)=1.
0008              IF(I.GE.M3D4) DORIG(I)=1.
0009              IF(I.GT.MD4.AND.I.LT.M3D4) DORIG(I)=-1.
0010           10 CONTINUE
0011              RETURN
0012              END
```

```
FORTRAN IV G LEVEL  21                 SYMEIG            DATE = 77062          06/12/54              PAGE 0001

0001          SUBROUTINE SYMEIG(NDIM,N,A,D,E,WANTX,X,AMULT,DMULT,XMULT)
0002          INTEGER NDIM,N
0003          LOGICAL WANTX
0004          REAL A(NDIM,NDIM),D(NDIM),E(NDIM),X(NDIM,NDIM)
        C
        CCOMPUTES EIGENVALUES AND EIGENVECTORS OF REAL SYMMETRIC MATRIX
        C
        C  NDIM = DECLARED ROW DIMENSION OF A (AND X).
        C  N = ORDER OF A
        C  A = N-BY-N MATRIX
        C     SYMMETRIC INPUT MATRIX.  ONLY DIAGONAL AND LOWER TRIANGLE
        C        USED.  IF POSSIBLE, PUT BIGGEST ELEMENTS IN UPPER
        C        LEFT CORNER.  UPPER TRIANGLE UNALTERED UNLESS  A  AND
        C        X  ARE SAME ARRAY.
        C  D = N-VECTOR.  OUTPUT. EIGENVALUES
        C  E = N-VECTOR.  USED FOR WORK SPACE
        C  WANTX = .TRUE. IF EIGENVECTORS DESIRED. .FALSE. IF NOT.
        C  X = N-BY-N MATRIX
        C     IF (WANTX) THEN OUTPUT X(*,J) IS EIGENVECTOR ASSOCIATED
        C     WITH EIGENVALUE D(J).
        C     TO SAVE STORAGE, X AND A MAY BE THE SAME ARRAY.  THUS
        C           CALL SYMEIG(NDIM,N,A,D,E,NONZERO,A)
        C     REPLACES INPUT MATRIX WITH ITS EIGENVECTORS.
        C
0005          REAL ALPHA,BETA,GAMMA,KAPPA,AIJ,T,C,S,F
0006          REAL ABS,SQRT
        C
        C     HOUSEHOLDER TRIDIAGONALIZATION
        C
0007          D(1) = A(1,1)
0008          IF (N.LE.2) GO TO 8
0009          NM1 = N-1
0010          DO 7 K = 2, NM1
        C
        C        FIND REFLECTION WHICH ZEROS A(I,K-1), I=K+1,....,N
        C
0011             ALPHA = 0.
0012             DO 1 I = K, N
0013                D(I) = A(I,K-1)
0014                ALPHA = ALPHA + D(I)**2
0015      1      CONTINUE
0016             ALPHA = SQRT(ALPHA)
0017             IF (D(K).LT.0.) ALPHA = -ALPHA
0018             D(K) = D(K) + ALPHA
0019             BETA = ALPHA * D(K)
0020             IF (BETA.EQ.0.) GO TO 6
        C
        C        APPLY REFLECTION TO BOTH ROWS AND COLUMNS OF REST OF A
        C        USE AND COMPUTE ONLY LOWER TRIANGLE
        C
0021             KAPPA = 0.
0022             DO 3 I = K, N
0023                GAMMA = 0.
```

```
               DO 2 J = K, N
                  IF (I.GE.J) AIJ = A(I,J)
                  IF (I.LT.J) AIJ = A(J,I)
                  GAMMA = GAMMA + AIJ * C(J)
      2        CONTINUE
               E(I) = GAMMA/BETA
               KAPPA = KAPPA + C(I) * E(I)
      3     CONTINUE
            KAPPA = C.5 * KAPPA / BETA
            DO 5 I = K, N
               E(I) = E(I) - KAPPA * D(I)
               DO 4 J = K, I
                  A(I,J) = A(I,J) - D(I)*E(J) - E(I)*D(J)
      4        CONTINUE
      5     CONTINUE
C
      6     A(K,K-1) = D(K)
            D(K) = A(K,K)
            A(K,K) = BETA
            E(K) = -ALPHA
      7 CCNTINUE
      8 C(N) = A(N,N)
          E(N) = A(N,N-1)
C
C         ACCUMULATE TRANSFORMATIONS
C         PRODUCES X SO THAT  XT*(INPUT A)*X  IS TRIDIAGONAL
C
          IF (.NOT.WANTX) GO TO 20
          X(N,N) = 1.
          DC 15 KB = 1, NM1
            K = N-KB
            BETA = A(K,K)
            DO 11 I = K, N
               X(I,K) = 0.
               X(K,I) = 0.
     11     CONTINUE
            X(K,K) = 1.
            IF (K.EQ.1) GO TO 15
            IF (BETA.EQ.0.) GO TO 15
            DO 14 J = K, N
               GAMMA = 0.
               DO 12 I = K, N
                  GAMMA = GAMMA + X(I,J) * A(I,K-1)
     12        CONTINUE
               GAMMA = GAMMA / BETA
               DO 13 I = K, N
                  X(I,J) = X(I,J) - GAMMA * A(I,K-1)
     13        CONTINUE
     14     CONTINUE
     15 CCNTINUE
C
C         TRIDIAGONAL QR ALGORITHM
C         IMPLICIT SHIFT FROM LOWER 2-BY-2
```

```
               C
 0069             20 CC 27 MB = 2, N
 0070                   M = N+2-MB
 0071                   MM1 = M-1
 0072                   ITER = C
 0073                   L = 1
 0074             21    E(L) = 0.
               C
               C        FIND L SUCH THAT E(L) IS NEGLIGIBLE
               C
 0075                   L = M
 0076             22    S = ABS(D(L-1)) + ABS(D(L))
 0077                   T = S + ABS(E(L))
 0078                   IF ( T .EQ. S ) GO TO 23
 0079                   L = L-1
 0080                   IF (L.GE.2) GO TO 22
               C
               C        IF E(M) IS NEGLIGIBLE, THEN D(M) IS AN EIGENVALUE, SO..
               C
 0081             23    IF (L.EQ.M) GO TO 27
 0082                   ITER = ITER + 1
               C
               C        FORM IMPLICIT SHIFT
               C
 0083                   T = (D(M-1) - D(M))/(E(M) + E(M))
 0084                   S = SQRT(1. + T*T)
 0085                   IF (T.LT.0.) S = -S
 0086                   E(L) = D(L) - D(M) + E(M)/(T+S)
 0087                   F = E(L+1)
               C
               C        CHASE NONZERO  F  DOWN MATRIX
               C
 0088                   DO 26 J = L, MM1
 0089                      T = ABS(E(J)) + ABS(F)
 0090                      ALPHA = T*SQRT((E(J)/T)**2 + (F/T)**2)
 0091                      C = E(J) / ALPHA
 0092                      S = F / ALPHA
 0093                      BETA = S*(D(J+1) - D(J)) + 2.*C*E(J+1)
 0094                      E(J) = ALPHA
 0095                      E(J+1) = E(J+1) - C*BETA
 0096                      T = S*BETA
 0097                      D(J) = D(J) + T
 0098                      D(J+1) = D(J+1) - T
 0099                      IF (J.EQ.MM1) GO TO 24
 0100                      F = S*E(J+2)
 0101                      E(J+2) = -C*E(J+2)
               C
 0102             24       IF (.NOT.WANTX) GO TO 26
 0103                      DO 25 I = 1, N
 0104                         T = X(I,J)
 0105                         X(I,J) = C*T + S*X(I,J+1)
 0106                         X(I,J+1) = S*T - C*X(I,J+1)
 0107             25       CONTINUE
```

```
             C
 0108           26     CCNTINUE
 01C9                  GO TO 21
             C
 0110           27 CCNTINUE
             C
             C     NORMALIZE RESULTS
             C
 0111              DMULT=0.
 0112              XMULT=0.
 0113              CC 29 I=1,N
 0114              CT=ABS(D(I))
 0115              IF(DMULT.LT.DT) DMULT=CT
 0116              DC 28 J=1,N
 0117              XT=ABS(X(I,J))
 0118              IF(XMULT.LT.XT) XMULT=XT
 0119           28 CCNTINUE
 0120           29 CCNTINUE
 0121              CC 31 I=1,N
 0122              D(I)=D(I)/DMULT
 0123              DO 30 J=1,N
 0124              IF(XMULT.EQ.0.) GO TO 30
 0125              X(I,J)=X(I,J)/XMULT
 0126           30 CCNTINUE
 0127           31 CCNTINUE
 0128              RETURN
 0129              END
```

BIBLIOGRAPHY

1. R. E. Bellman, Introduction to Matrix Analysis, McGraw Hill, New York, pp. 234-235, 1960.

2. T. L. Boullion, P. L. Odell, Generalized Inverse Matrices, Wiley-Interscience, 1971.

3. E. Oran Brigham, The FFT, Englewood Cliffs, N.J., Prentice-Hall, Inc., 1974.

4. W. L. Brogan, Modern Control Theory, Quantum Publishers, Inc., Park Ave. South, New York, Chapter 4-6, 1974.

5. A. B. Campbell, "A New Sampling Theorem for Causal (Non-Bandlimited) Functions," Ph.D. Dissertation, University of New Mexico, Electrical Engineering and Computer Science Department, 1973.

6. R. E. Crochiere and L. R. Rabiner, "Optimum FIR Digital Filter Implementations for Decimation, Interpolation, and Narrow-Band Filtering." IEEE Trans. on Acoustics, Speech, and Signal Processing, Vol. ASSP-23, No. 5, pp. 444-456, Oct. 1975.

7. C. G. Cullen, Matrices and Linear Transformations, Addison-Wesley, Reading, Mass., pp. 106-108, 1967.

8. G. Dahlquist and A. Bjorck, Numerical Methods, Translated by N. Anderson, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, pp. 81-135, 405-421, 1974.

9. J. G. F. Francis, "The QR Transformation," Computer Journal, Vol. 4, pp. 265-271, 332-345, 1961-1962.

10. J. N. Franklin, Matrix Theory, Prentice Hall, Inc., Englewood Cliffs, New Jersey, pp. 1-121, 1968.

11. W. Fulks, Advanced Calculus, John Wiley and Sons, Inc., New York, pp. 61-62, 1961.

12. R. T. Gregory and D. L. Karney, A Collection of Matrices for Testing Computational Algorithms, John Wiley and Sons, New York, pp. 22-23, 1969.

13. Robert M. Grey, "Toeplitz and Circulant Matrices: A Review," Stanford University, Information Systems Laboratory, Technical Report No. 6502-1, pp. 16-21, June 1971.

14. R. V. L. Hartley, "The Transmission of Information," Bell System Technical Journal, Vol. 7, pp. 535-560, 1928.

15. B. R. Hunt, "A Matrix Theory Proof of the Discrete Convolution Theorem," IEEE Transactions on Audio and Electroacoustics, Vol. AU-19, No. 4, pp. 285-288, Dec. 1971.

16. P. Lancaster, Theory of Matrices, New York: Academic Press, 1969.

17. K. S. Lin, "A Digital Signal Processing Approach to Interpolation Filters." Ph.D. Dissertation, University of New Mexico, Department of Electrical Engineering and Computer Science, 1976.

18. V. Mangulis, Handbook of Series for Scientists and Engineers, Academic Press, New York, p. 20, 1965.

19. J. McNamee, et al, "Whittaker's Cardinal Function in Retrospect," Mathematics of Computation, Vol. 25, Number 113, pp. 141-154, January 1971.

20. C. Moler, Matrix Eigenvalues and Least Squares Computations, Unpublished Book, University of New Mexico, Chapter 7, Draft February 1973.

21. H. Nyquist, "Certain Topics in Telegraph Transmission Theory," Transactions, American Institute of Electrical Engineers, Vol. 47, pp. 617-644, 1928.

22. G. Oetken, T. Parks, H. Schüssler, "New Results in the Design of Digital Interpolators, IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. ASSP-23, pp. 301-309, June 1975.

23. A. Oppenheim, R. Schafer, Digital Signal Processing, Prentice Hall, Inc., Englewood Cliffs, New Jersey, Chapter 3, 1975.

24. A. C. Paley, N. Wiener, "Fourier Transformations in the Complex Domain," American Mathematical Society Colloquium Publication 19, New York, 1934.

25. A. Papoulis, The Fourier Integral and Its Applications, McGraw Hill Book Co., Inc., New York, Chapter 10, 1962.

26. L. R. Rabiner and R. C. Crochiere, "A Novel Implementation for Narrow Band FIR Digital Filter," IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. ASSP-23, No. 5, Oct. 1975.

27. R. W. Schaefer and L. R. Rabiner, "A Digital Signal Processing Approach to Interpolation," Proc, IEEE, Vol. 61, pp. 692-702, June 1973.

28. F. Scheid, "Numerical Analysis," Schaum's Outline Series, McGraw Hill Book Co., New York, Chapters 2-6, 1968.

29. C. E. Shannon, "Communication in the Presence of Noise," Proc. of the Institute of Radio Engineers, Vol. 37, pp. 10-21, January 1949.

30. C. E. Shannon, "A Mathematical Theory of Communication," Bell System Technical Journal, Vol. 27, pp. 379-423, 623-656, 1948.

31. W. D. Stanley, Digital Signal Processing, Prentice-Hall Book Company, Virginia, Chapter 3, 1975.

32. S. D. Stearns, Digital Signal Analysis, Hayden Book Co., Inc., 1975.

33. G. W. Stewart, Introduction to Matrix Computations, Academic Press, New York, Chapter 4, 1973.

34. F. Theilheider, "A Matrix Version of the FFT," IEEE Transactions Audio and Electroacoustics, Vol. AU-17, pp. 158-161, June 1969.

35. H. Urkowitz, "Parallel Realization of Digital Interpolation Filters for Increasing the Sampling Rate," IEEE Trans. on Circuits and Systems, Vol. CAS-22, No. 2, pp. 146-154, Jan. 1975.

36. E. T. Whittaker, "On the Functions Which are Represented by the Expansions of the Interpolation Theory," Proceedings of the Royal Society of Edinburg, Vol. 35, pp. 181-194, 1915.

37. J. M. Whittaker, "On the Cardinal Function of Interpolation Theory," Proceedings, Edinburg Mathematical Society, Ser. 2 Vol. 1, pp. 41-46, 1927.

38. J. M. Whittaker, Interpolatory Function Theory, Cambridge, London, 1935.

39. J. H. Wilkinson, The Algebraic Eigenvalue Problem, Clarendon Press: Oxford, 1965.

40. J. H. Wilkinson, "Global Convergence of Tridiagonal QR Algorithm with Origin Shifts," Linear Algebra and Its Applications, Vol. 1, pp. 409-420, 1968.

41. N. P. Zhidov, I. S. Berezin, Computing Methods, Vol. 1, Pergamon Press, p. 123, 1965.

CURRICULUM VITAE

Joseph Charles Wheeler, the son of John Austin and Dottie Thompson Wheeler, was born on September 11, 1939 in Caddo Parish, Shreveport, Louisiana. He received his primary and secondary education in Shreveport, and graduated from Fair Park High School in 1957. In 1961, he graduated from Texas A and M College, at College Station, and immediately entered active commissioned duty in the United States Air Force.

Major Wheeler's professional education began with the MSEE degree from the Air Force Institute of Technology (AFIT) in Dayton, Ohio, 1964-65, followed by Squadron Officers School at Montgomery, Alabama, 1967, and the Air Command and Staff College Seminar at LG Hanscom Field, MA, 1972-74. In July 1974 he was selected to attend the University of New Mexico under the AFIT Civilian Institution Program.

Major Wheeler's military assignments have been in Radar, Air Traffic and Command and Control, Electronic-Counter Measures, and Geophysical Data Processing. Upon completion of the Doctoral studies, he was promoted to Lieutenant Colonel and assigned to Headquarters United States Air Force at the Pentagon.

He is a Registered Professional Engineer, a member of the Eta Kappa Nu Association, the Society of Sigma Xi, and Phi Kappa Phi.