Analysis of Twinning using Fourier and Wavelet Transforms

A Thesis
Presented for the
Master of Science
Degree

The University of Tennessee, Knoxville

William Dean Turley
December 2002

Dedication

This thesis is dedicated to my parents Junious and Helen Turley.

Acknowledgments

I would like to express my gratitude to Dr. Charles Collins for his guidance and patience. I would also like to thank members of my supervisory committee, Dr. Samuel Jordan and Dr. Suzanne Lenhart, for the time spent reviewing my work and the helpful suggestions for revision. I would like to thank my father, Junious Turley, for his encouragement. Finally, I would like to thank Toby Koosman and my brother Tim Turley for their support and a handy computer.

Abstract

In previous research on twinning, Finite Element Methods were used to examine the relationship between twinning microstructure, boundary conditions imposed on the twinning model, and a measure of the success of twinning; the Young measure. Minimizing sequences of deformations for the energy functional were used to approximate twinning microstructure. Since twinned material has a banded microstructure and the uniformity of the bands is related to successful twinning, we investigate a method of measuring twinning quality using Fourier and wavelet transforms. Fourier transforms detect global qualities, such as the frequency of twinning bands in this case, while wavelet transforms capture local qualities well, such as phase changes in twinning bands. We produce a measure of twinning quality from statistical analysis of data obtained from Fourier and wavelet transforms. The quality number distinguishes between twinning with uniform banded patterns and twinning without uniform banded patterns.

Table of Contents

Introduction	1
1. Twinning Model	3
2. Fourier Series and Fourier Transforms	7
2.1 Fourier Series	7
2.2 Discrete Fourier Transform	10
2.3 Discrete Fourier Transform Examples	11
3. Wavelets	22
3.1 Haar Wavelets	22
3.2 Daubechie Wavelets	27
3.3 Discrete Wavelet Transform	31
3.4 Wavelet Transform Examples	33
4. Twinning Analysis	38
4.1 Introduction	38
4.2 Data Conditioning Algorithms	40
4.3 Frequency Detection	47
4.4 DWT Analysis	53
4.5 Quality Numbers	55
4.6 Results	58
Conclusion	67
Bibliography	68
Appendix	70
Vita	79

List of Figures

2.3.1 DFT of $f(n) = 0.2 \cdot \sin(2\pi 6x_n) + 0.2 \cdot \sin(2\pi 18x_n)$. 13

2.3.2 DFT coefficients of the delta function. 14

2.3.3 DFT of $f(n) = 0.2 \cdot \sin(2\pi 6x_n) + 0.2 \cdot \sin(2\pi 18x_n)$. 15

2.3.4 DFT coefficients of transient signal. 16

2.3.5 DFT of $f(n) = 0.2 \cdot \sin(2\pi 6.518x_n)$. 17

2.3.6 DFT of $f(n) = x_n + \frac{1}{10}\sin(2\pi 10x_n)$. 18

2.3.7 DFT of $f(n) = 0.1 \cdot r_n + 0.1 \cdot \sin(2\pi 10x_n)$ 19

2.3.8 DFT of $f(n) = x_n + 0.1 \cdot r_n + 0.1 \cdot \sin(2\pi 10x_n)$. 20

2.3.9 DFT of $f(n) = t(6x_n) + t(18x_n)$. 21

3.1.1 Haar scaling function 24

3.1.2 Haar basis function 25

3.1.3 Haar basis functions $\psi_{1,0}$ and $\psi_{1,1}$ 26

3.2.1 D-4 scaling function 28

3.2.2 D-4 basis function 28

3.3.1 Mallat's algorithm 32

3.4.1 DWT coefficients for delta function 34

3.4.2 DWT coefficients for linear function 35

3.4.3 DWT coefficients for linear function and sine function 36

3.4.4 DWT coefficients for transient function 37

4.2.1 Plot of $u1$ verses $u2$ 43

4.2.2 Plot of *v*1 verses *v*2 44

4.2.3 Plot of *w*1 verses *w*2 45

4.2.4 Plot of row of *w*1 verses row of *w*2 data 46

4.3.1 Plot of triangle wave fit to data 52

4.4.1 Plot of DWT coefficients showing phasechange 54

4.6.1 Twinning data Q = 100 61

4.6.2 Twinning data Q = 95 61

4.6.3 Twinning data Q = 39 62

4.6.4 Twinning data Q = 64 62

4.6.5 Typical row data from Figure 4.6.4 63

4.6.6 Twinning data Q = 99 64

4.6.7 Twinning data Q = 56 64

4.6.8 Twinning data Q = 31 65

4.6.9 Twinning data Q = 52 65

4.6.10 Twinning data Q = 82 66

Introduction

Twinning is a special orientation between crystals in a solid material. In twinned material we can identify bands in the solid where crystals have a special orientation. In previous research [Collins, 1990] numerical methods were used to compute approximations of twinning. These approximations minimize an energy functional and were used to test the feasibility of twinning for a material. In this thesis we analyze a set of approximations by using Fourier and wavelet transforms to measure their quality.

Chapter 1 starts with a discussion of the mathematical model for twinning in order to develop notation for the data analysis that follows. We begin with a microscopic description of twinning using lattice vectors and then describe how this is linked by the Born hypothesis to the macroscopic or continuum description of twinning. We describe how the Finite Element Method is used and show how this sets up much of the notation that is used in our analysis.

Chapter 2 is about Fourier transforms and properties of the discrete Fourier transform. First, we discuss the theory of Fourier transforms to motivate their use and application to twinning. Examples are used to illustrate the results given by the Fourier transform applied to data that represents twinning.

The theory and application of wavelet transforms is covered in Chapter 3, again with emphasis on results related to analyzing twinning data. This transform provides good localized information about data, but it does not give very precise frequency measurements. The results of using the wavelet transform on test data are given and discussed.

Finally, the methods of twinning analysis are covered in Chapter 4. The twinning data we analyze is similar to a triangle wave so it was necessary to develop a filter for the Discrete Fourier Transform to find the frequency of this type of data. The computational results of the filter are given for test data in the form of triangle wave with different amplitudes, frequencies and phase shifts that are within the bounds of the data encountered. Wavelet analysis was used to detect phase changes in twinning data. We analyzed statistics from Fourier and wavelet analysis for our final results.

Chapter 1

Twinning Model

To develop notation for our data analysis, we start with a description of how the data was generated. First we need to understand the underlying process of twinning. The data represents twinned material where twinning is identified by planes of discontinuity in crystalline material. To describe twinning, two sets of lattice vectors, (e_i) and $(\bar{e}_i)$, are assigned to opposite sides of the plane of discontinuity so that $(\bar{e}_i) = R(e_i)$ for some $R \in \vartheta^+$, the group of rotations. The lattice vectors conform to two more properties: (1) There exists a set of vectors $(\tilde{e}_i)$ opposite the plane of discontinuity from the set of vectors (e_i) such that the normal to the plane of discontinuity can be written as $n = n_i e_i$ and as $n = n_i \tilde{e}_i$ with the same values of n_i; (2) There exists a set of lattice vectors $(\hat{e}_i)$ opposite the plane of discontinuity from the set of lattice vectors (e_i) such that for all vectors v in the plane, i.e. where $v \cdot n = 0$, we have $v = v_i e_i = v_i \hat{e}_i$ with the same values of v_i. For ease we assume that $\hat{e}_i = \tilde{e}_i$. From these assumptions $\hat{e}_i = (I + a \otimes n) e_i$ for some amplitude vector a. In addition, since the lattice is the same on each side up to a rotation $\det[I + a \otimes n] = 1$, implying $a \cdot n = 0$. Let the crystal symmetry group

$$G = \{(\mu_i^j) : \mu_i^j \in Z, \det \mu = \pm 1\}$$

be the set of all transformations giving an equivalent lattice. In addition, we associate the conjugate group $\overline{G} = \{H : He_i = \mu_i^j e_j \text{ for some } \mu \in G\}$ with a given set of lattice vectors which, with further restrictions, will give us Laue groups. Laue groups allow us to

confine our analysis to crystal symmetries unique to this problem. Now, since $\bar{e}_i = \mu_i^j \hat{e}_i$ for some $\mu \in G$ and $\bar{e}_i = \mathrm{Re}_i$ we have $\mathrm{Re}_i = \mu_i^j (I + a \otimes n) e_j$. Then the twinning equation can be written as

$$R = (I + a \otimes n) H, \quad He_i = \mu_i^j e_j .$$

Twinning is possible if given $H \in \overline{G}$, this equation can be solved for R, a and n.

The lattice vector representation of crystals that works on a microscopic scale and leads to the twinning equation above is linked to a continuum description on a macroscopic scale by the Born hypothesis. Assume we have a reference domain, $\Omega \in \Re^3$, with reference lattice vectors $\left(e_i^0\right)$ such that a lattice, $L\left(e_i^0\right)$, is identified with the reference domain. Suppose a property depends on lattice vectors and $u : \Omega \to \Re^3$. The Born hypothesis states that the value of the property at a point $u(x)$ is the value of the property at the reference lattice vectors as deformed by $\nabla u(x)$, i.e. $e_i = \nabla u(x) e_i^0$. An equivalent form for the twinning equation that is a consequence of the Born hypothesis is

$$RU = (I + a \otimes n) UH ,$$

where R,U and H are matrices, U is a deformation gradient and $H \in \overline{G}$ [Collins, dissertation].

Deformations correspond to a tendency of crystals in the twinned material to develop and orient themselves in the most energy- conserving configuration possible. Thus the bulk energy of the material is an important measure. The formula for bulk energy uses a function of the deformation gradient $F = \nabla u$ and temperature θ as given by $J(u) = \int_\Omega W(\nabla u(x), \theta) dx$. There are several properties for the strain energy density,

$W(F,\theta)$, that have consequences for the mathematical model that has been developed for this physical problem. $W(F,\theta)$ has invariance of energy with respect to orientation so $W(RF)=W(F)$ for $R\in\mathcal{O}^+$. $W(F,\theta)$ has invariance due to crystal symmetry so the energy satisfies $W(FH)=W(F)$ for all $H\in P$ the group of rotations that are allowed for a particular problem, and finally, $W(F)\geq 0$ but $W(F)=0$ only if $F=RUH$ where U is the deformation gradient associated with the diffusionless phase transition. Such an F is called a variant. In this paper we will be working with two wells, represented by the two gradients U_1 and U_2, and their variants.

The model that was developed for twinning involves minimizing the energy functional $J(u)$, i.e., find $u\in A$ such that $J(u)\leq J(v)$, $\forall v\in A$ where

$$A=\left\{v\in W^{1,\infty}(\Omega,\Re^3)\,|\,v=v_0(x)\ for\ x\in\partial\Omega\right\}.$$

and v_0 specifies the boundary conditions. We use the space $W^{1,\infty}$ to allow deformations that are continuous but can have discontinuous gradients. With some boundary conditions it is impossible to find $u\in A$ with $J(u)=0$, however it is possible to find a sequence $\{u_k\}\subset A$ such that $\lim_{k\to\infty}J(u_k)=0$. These correspond to twinned deformations. In addition, since $\lim_{k\to\infty}J(u_k)=0$ but $J\left(\lim_{k\to\infty}u_k\right)>0$ we have minimizing sequences of functions but not a minimizing function for $J(u)$.

One way to produce such a sequence, is to use the Finite Element Method to approximate solutions on different mesh sizes. In particular we work in two dimensions and take the reference domain to be $\Omega=[0,1]^2$. Then we triangulate Ω with a uniform mesh of size $h_N=1/N$, $N\in Z^+$. Each coordinate is determined by an integral multiple of

the mesh size so $x_i = ih_N$ and $y_j = jh_N$ for $i, j = 0,1,2,...N$. The boundary condition v_0 is defined by $v_0(x, y) = F\begin{pmatrix} x \\ y \end{pmatrix}$ where F is a given 2×2 matrix. The Finite Element Approximation then produces values $\{u(x_i, y_j): 0 \le i, j \le N\}$. Our goal is to analyze this data to determine twinning patterns, detect defects and finally to produce a quality measure.

Chapter 2

Fourier Series and Fourier Transforms

Section 2.1

Fourier Series

The Fourier series and the discrete Fourier transform convert functions of time or space to functions of frequency. This conversion allows us characterize the oscillatory behavior that we expect to find in our data. In this section, we discuss Fourier series in the real and complex forms, and in Section 2.2, we discuss the Discrete Fourier Transform. The real variable definition of Fourier series will allow us to provide clear examples for the problems we expect to encounter. The complex variable definition of Fourier series allows us to outline a clear analogue between Fourier series and the finite sum form that is used in the discrete Fourier transform.

Fourier series are linear combinations of trigonometric functions. On the interval $[-\pi,\pi]$, the set of real functions is

$$\left\{(2\pi)^{-\frac{1}{2}},\left(\pi^{-\frac{1}{2}}\cos x\right),\left(\pi^{-\frac{1}{2}}\sin x\right),\ldots,\left(\pi^{-\frac{1}{2}}\cos nx\right),\left(\pi^{-\frac{1}{2}}\sin nx\right),\ldots\right\}$$

while the set of complex functions is $\left\{(2\pi)^{-\frac{1}{2}}e^{inx}\right\}_{n=-\infty}^{\infty}$. Since these functions are orthonormal, we have [Rudin, p.187]

$$\frac{1}{\pi\sqrt{2}}\int_{-\pi}^{\pi}\cos nx=\frac{1}{\pi\sqrt{2}}\int_{-\pi}^{\pi}\sin nx=0 \quad for\ n=1,2,3,\ldots$$

$$\frac{1}{\pi}\int_{-\pi}^{\pi}\cos(mx)\cos(nx)dx=\begin{cases}1 & if\ m=n\\ 0 & if\ m\neq n\end{cases}$$

$$\frac{1}{\pi}\int_{-\pi}^{\pi}\sin(mx)\sin(nx)dx=\begin{cases}1 & if\ m=n\\ 0 & if\ m\neq n\end{cases}$$

$$\int_{-\pi}^{\pi}\sin(mx)\cos(nx)dx=0 \quad for\ m,n=1,2,3,...$$

The basis functions for the real form of the Fourier transform are related to the exponential function in the complex Fourier transform by Euler's formula $e^{ix}=\cos x+i\sin x$. Thus, the basis functions $\sin(mx)$ and $\cos(nx)$ are implicit in the complex Fourier transform. For positive integers *m* and *n* we have for the set of complex functions with $\phi_n(x)=(2\pi)^{-\frac{1}{2}}e^{inx}$,

$$\frac{1}{2\pi}\int_{-\pi}^{\pi}\phi_m\bar{\phi}_n dx=\begin{cases}1 & if\ m=n\\ 0 & if\ m\neq n\end{cases}. \qquad \textbf{(2.1.1)}$$

For these sets of functions we define the Fourier coefficients for a function $f(x)$ that is integrable on $[-\pi,\pi]$ as

$$a_n(f)=\frac{1}{\sqrt{\pi}}\int_{-\pi}^{\pi}f(x)\cos(nx)dx,\ \ n=0,1,2,...$$

$$b_n(f)=\frac{1}{\sqrt{\pi}}\int_{-\pi}^{\pi}f(x)\sin(nx)dx,\ \ n=1,2,....$$

In the complex form we have

$$c_n(f)=\int_{-\pi}^{\pi}f(x)\bar{\phi}_n(x)dx,\ \ n=-\infty,...,\infty \qquad \textbf{(2.1.2)}$$

The real form of a Fourier series representation of $f(x)$ is [Davis p.86]

$$f(x)=\frac{a_0}{2}+\sum_{n=1}^{\infty}a_n\cos(nx)+b_n\sin(nx)$$

The complex Fourier series is [Rudin p. 187]

$$f(x)=\sum_{n=-\infty}^{\infty}c_n\phi_n(x). \qquad \textbf{(2.1.3)}$$

There is symmetry in the values of the coefficients if $f(x)$ is a real function.

$\sum_{n=-N}^{N} c_n \phi_n(x)$ is a real function if and only if $c_{-n} = \overline{c}_n$ for $n = 0,1,...,N$. **(2.1.4)**

We will conclude this section with Parseval's Theorem and a discussion of the completeness of the space spanned by trigonometric functions. First we need several definitions. Let X be a measurable set. If a complex function f is defined on X and $\left\{\int_X |f|^2 dx\right\}^{\frac{1}{2}} < \infty$ then we say $f \in L^2(X)$. The distance between two functions f and g in $L^2(X)$ is defined to be $\|f-g\|_{L^2(X)} = \left\{\int_X |f-g|^2 dx\right\}^{\frac{1}{2}}$. We say that $\{f_n\}$ converges to f in L^2 if $\|f_n - f\|_{L^2(X)} \to 0$. The sequence $\{f_n\}$ is a Cauchy sequence in $L^2(X)$ if for every $\varepsilon > 0$ there is an integer N such that $n \geq N$, $m \geq N$ implies $\|f_n - f_m\|_{L^2(X)} \leq \varepsilon$. [Rudin p. 328]

Parseval's Theorem Let $f \in L^2([-\pi,\pi])$ and let $s_N = \sum_{n=-N}^{N} c_n \phi_n(x)$ be the N^{th} partial sum of the series $\sum_{-\infty}^{\infty} c_n \phi_n(x)$. Then

$$\lim_{N\to\infty} \|f - s_N\|_{L^2} = 0,$$

$$\sum_{-\infty}^{\infty} |c_n|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |f|^2 dx.$$

[Rudin p.328] In other words, the space of trigonmetric functions is complete and the l^2 norm of the coefficients c_n is equal to the L^2 norm of f.

Section 2.2

Discrete Fourier Transform

When a function is represented by values at a discrete set of points, we can still study the oscillations of the function. The discrete Fourier transform uses e^{-inx} that allows us to discuss our results in the context of the real Fourier transform basis functions $\sin(mx)$ and $\cos(nx)$ by using Euler's formula, $e^{-inx} = \cos(nx) - i\sin(nx)$. We use the discrete Fourier transform to quantify oscillations in the function f, using the value $\{f(n): n = 1,..., N\}$ where $f(n) = f(x_n)$ and $x_n = \frac{n-1}{N}$. We will denote the discrete Fourier transform with the abbreviation DFT.

DFT coefficients are evaluated with $c_k(f) = \sum_{n=1}^{N} f(n) e^{-2\pi i(k-1)(n-1)/N}$ for $k = 1, 2,..., N$, where N is the length of the data array. The i in the exponent of the exponential function represents $\sqrt{-1}$.

We also have an inverse DFT that has the form [Cartwright, p. 202]

$$f(n) = \left(\frac{1}{N}\right)\sum_{k=1}^{N} c_k e^{2\pi i(k-1)(n-1)/N} \text{ for } n = 1, 2,..., N. \qquad \textbf{(2.2.1)}$$

Parseval's theorem for the DFT states that for the set of coefficients $\{c_k\}$ obtained from the DFT of $f(n)$, we have $\sum_{k=1}^{N} |c_k(f)|^2 = \sum_{n=1}^{N} |f(n)|^2$.

Section 2.3

Discrete Fourier Transform Examples

In this section we give examples of DFT analysis of data. We fix the length of f to $N = 64$. Besides sine and cosine, two other functions we use in our examples are triangle waves and random numbers. Triangle waves are denoted by $t(\lambda x_n)$ where λ is the frequency in the interval $[0,1]$. Random numbers uniformly distributed in the interval $(0,1)$ are represented by $r_n = r(x_n)$. For data represented by $f(n)$, the coefficients from the DFT of f will be denoted by $c_k(f)$ for $k = 1,2,...,64$. Basis functions for the real form of the Fourier transform are related to the exponential function used in the DFT by Euler's formula, so we will discuss our results in terms of $\cos(nx)$ and $\sin(nx)$. The relation between the index k in $c_k(f)$ and frequency λ is $\lambda = k - 1$. Finally, by (2.1.4), $c_k(f) = \overline{c}_{66-k}(f)$ for all k if and only if f is real. Since $f(n)$ is real we limit our discussion to the first thirty-three coefficients.

The DFT is well suited to detecting the frequency of data. The graph of $|c_k|$ versus k from the DFT of f, where $f(n) = 0.2\sin(2\pi 6x_n) + 0.2\sin(2\pi 18x_n)$ is in Figure 2.3.1. In this example c_7 and c_{19} are nonzero and show that two frequencies are found: 6 and 18 cycles in the interval $[0,1]$. In contrast, Figure 2.3.2 shows that the DFT of a delta function has coefficients with $|c_k| = 1$ for all k. In this example the delta function $f(16) = 1$ while $f(n) = 0$ for all other entries. The examples given in Figures 2.3.3

through 2.3.8 illustrate the difficulties encountered when attempting to detect a dominant frequency; a frequency cannot always be detected by comparing values of $|c_k(f)|$.

In Figure 2.3.3, $f(n) = 0.2\sin(2\pi 6x_n) + 0.2\sin(2\pi 18x_n)$ with $f(57) = 0$ changed to $f(57) = 0.5$. Compare to Figure 2.3.1 where $c_7 = 6.4$ and $c_{19} = 6.4$. In Figure 2.3.3, however, $c_7 = 6.9$ and $c_{19} = 5.9$. One changed entry in $f(n)$ caused this change in DFT coefficients.

At the top of Figure 2.3.4 is the graph of $f(n) = 0.2\sin(2\pi 6x_n)$ replaced with $g(n) = 0.2\sin(2\pi 18x_n)$ for $n = 17, \ldots, 29$. At the bottom are the DFT coefficients that show c_7 is largest in magnitude, but several coefficients have values near $|c_{19}|$ although the coefficient associated with $\lambda = 18$ for g is c_{19}.

In Figure 2.3.5 we have $|c_k|$ from the DFT of $f(n) = 0.2\sin(2\pi 6.518x_n)$. The DFT gives only integer values of frequency so we obtain competing values of $|c_k|$ when a signal contains a non-integer frequency. This limits precision when frequency is identified only with $|c_k|$. In this case $|c_7| = 4.0775$ and $|c_8| = 4.0809$.

In Figure 2.3.6 we have the DFT of $f(n) = x_n + \frac{1}{10}\sin(2\pi 10x_n)$. This graph shows how a linear term influences $|c_k|$. In this case $|c_1|$ through $|c_4|$ clearly dominate the coefficient $|c_{11}|$ associated with $\lambda = 10$. In twinning data, linear trends were removed so accurate frequency detection was possible.

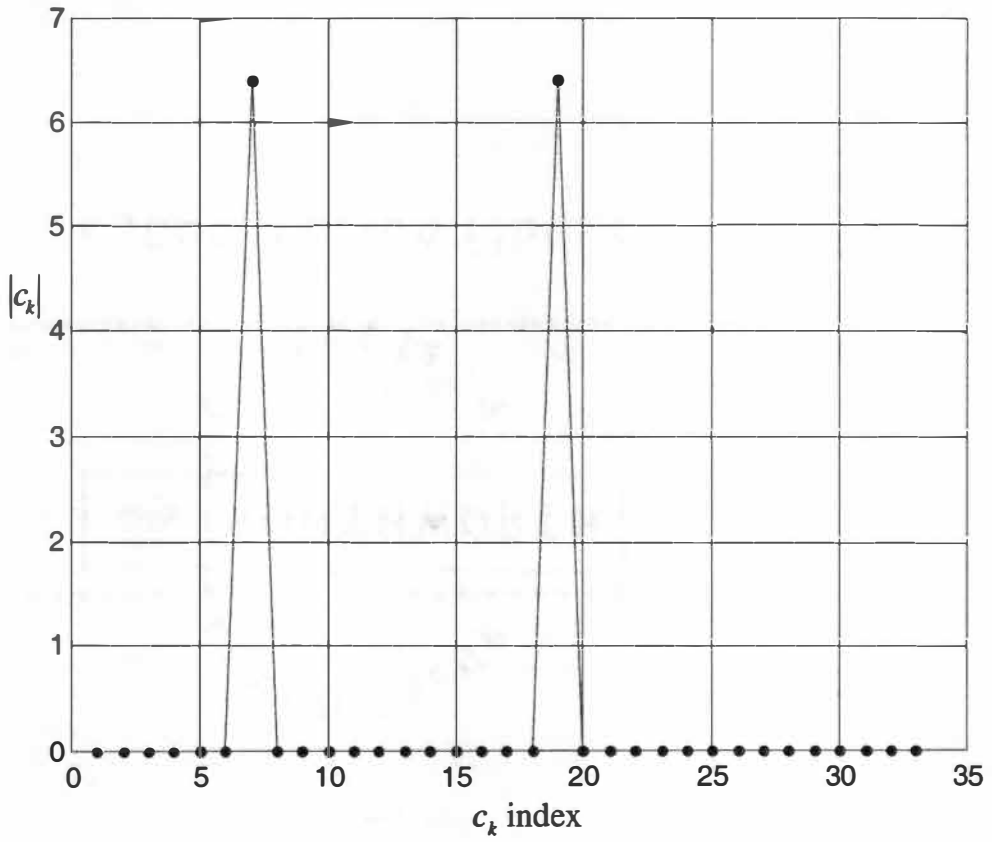

Figure 2.3.1 DFT of $f(n) = 0.2 \cdot \sin(2\pi 6x_n) + 0.2 \cdot \sin(2\pi 18x_n)$.

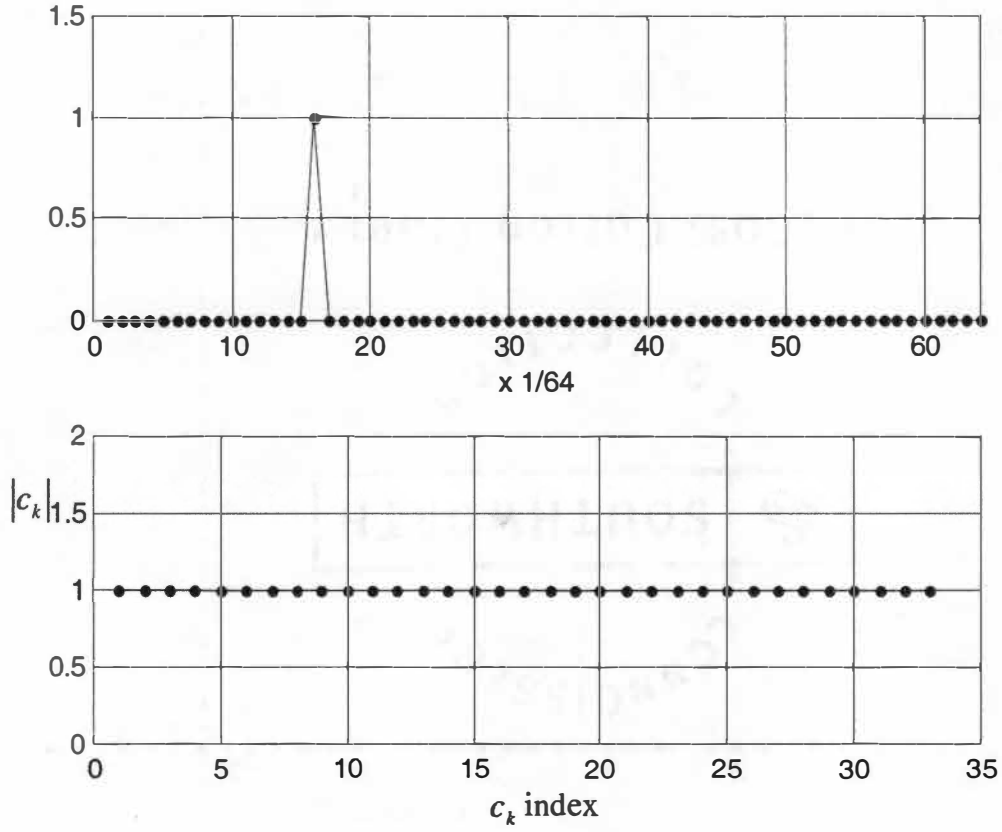

Figure 2.3.2 Delta function (top), DFT coefficients of the delta function (bottom).

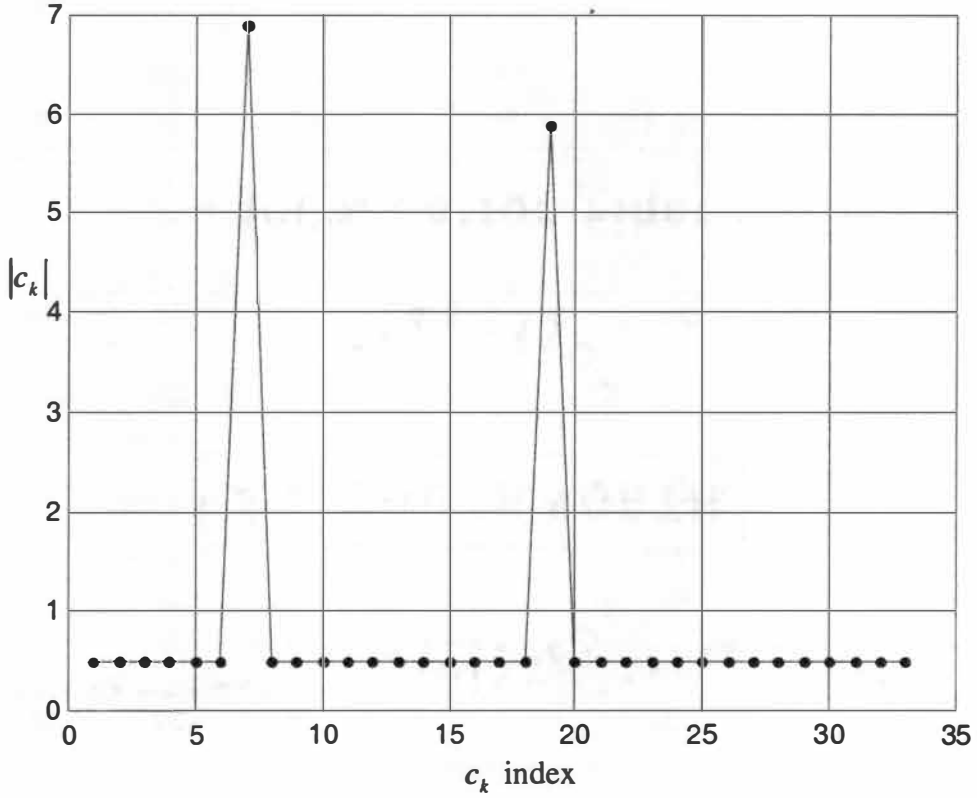

Figure 2.3.3 DFT of $f(n) = 0.2 \cdot \sin(2\pi 6x_n) + 0.2 \cdot \sin(2\pi 18x_n)$.

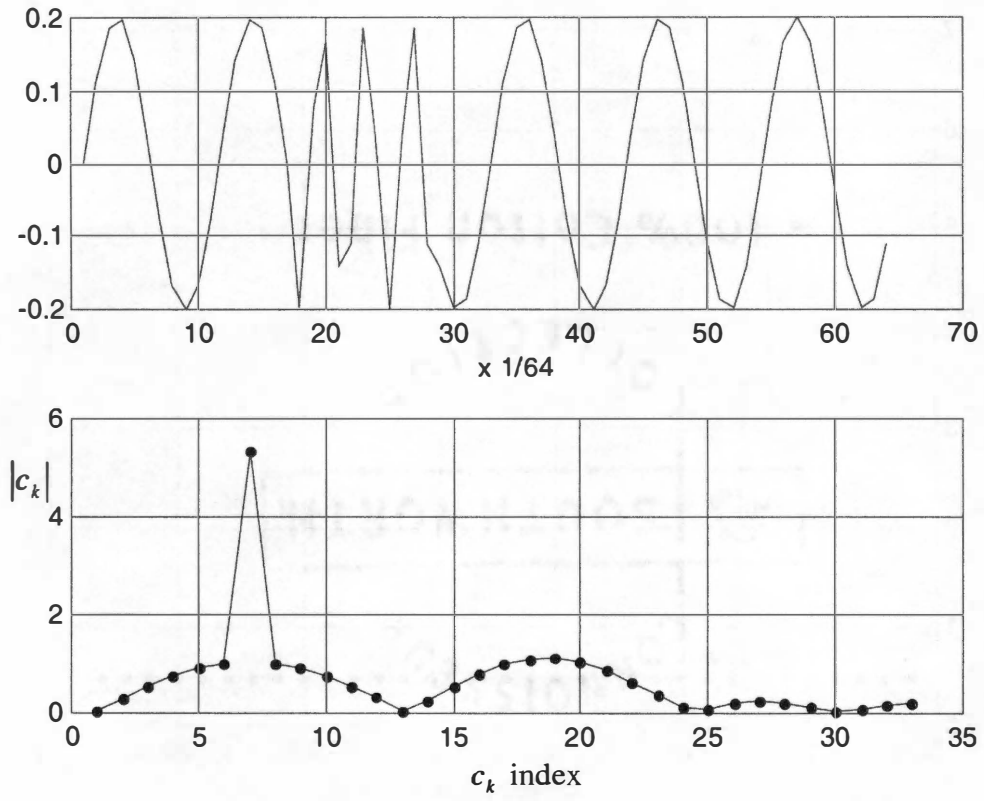

Figure 2.3.4 Graph of $f(n) = 0.2 \cdot \sin(2\pi 6x_n)$ with transient $g(n) = 0.2\sin\cdot(2\pi 18x_n)$ (top), DFT coefficients of transient (bottom).

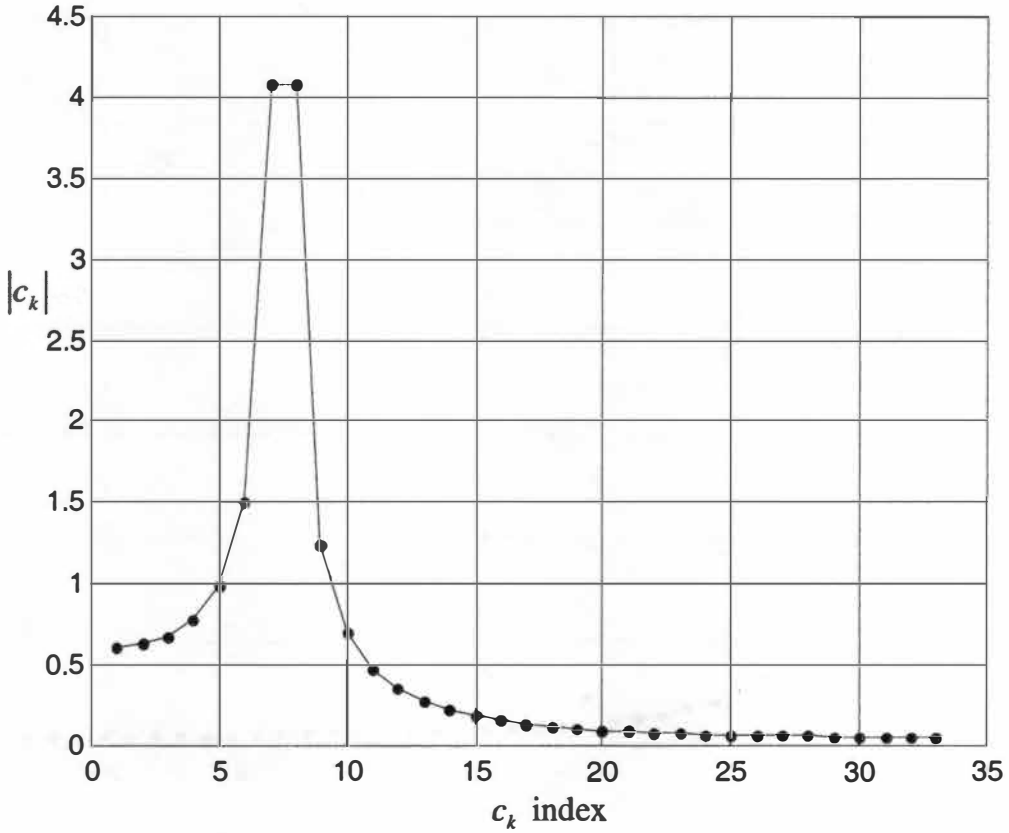

Figure 2.3.5 DFT of $f(n) = 0.2 \cdot \sin(2\pi 6.518 x_n)$.

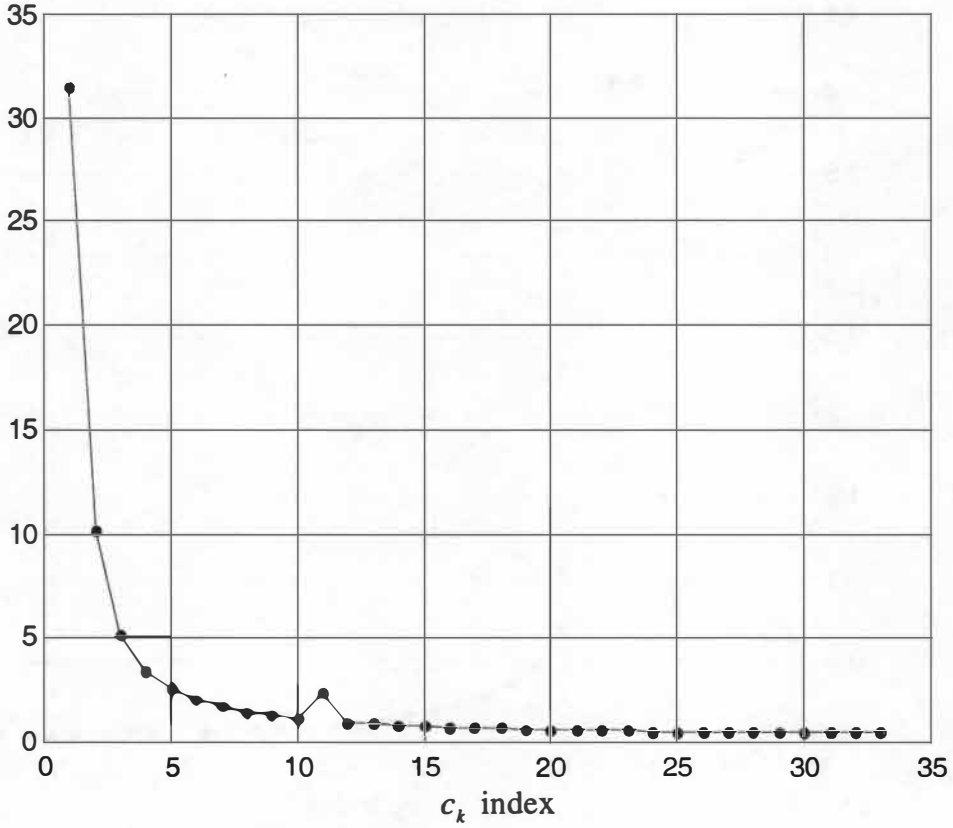

Figure 2.3.6 DFT of $f(n) = x_n + \frac{1}{10}\sin(2\pi 10 x_n)$.

In Figure 2.3.7 the DFT of $f(n) = 0.1 \cdot r_n + 0.1 \cdot \sin(2\pi 10 x_n)$. Without the random noise only $|c_{11}| \neq 0$ but we have $|c_k| > 0 \ \forall n$ with $|c_1| > |c_{11}|$. This is another example that the frequency of a signal is not directly related to the magnitude of the $|c_k|$'s.

Figure 2.3.8 is a graph of the DFT of $f(n)$ with linear trends and noise. Again, $\max\{|c_k|\}$ is not associated with $\lambda = 10$.

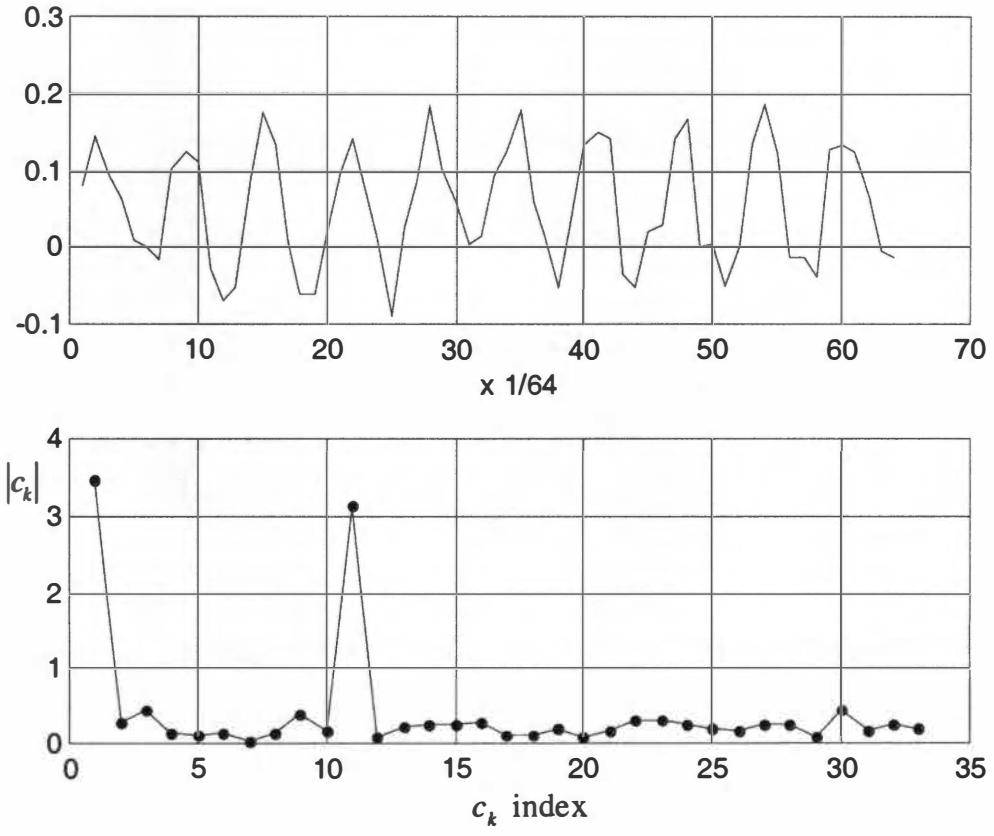

Figure 2.3.7 DFT of $f(n) = 0.1 \cdot r_n + 0.1 \cdot \sin(2\pi 10 x_n)$

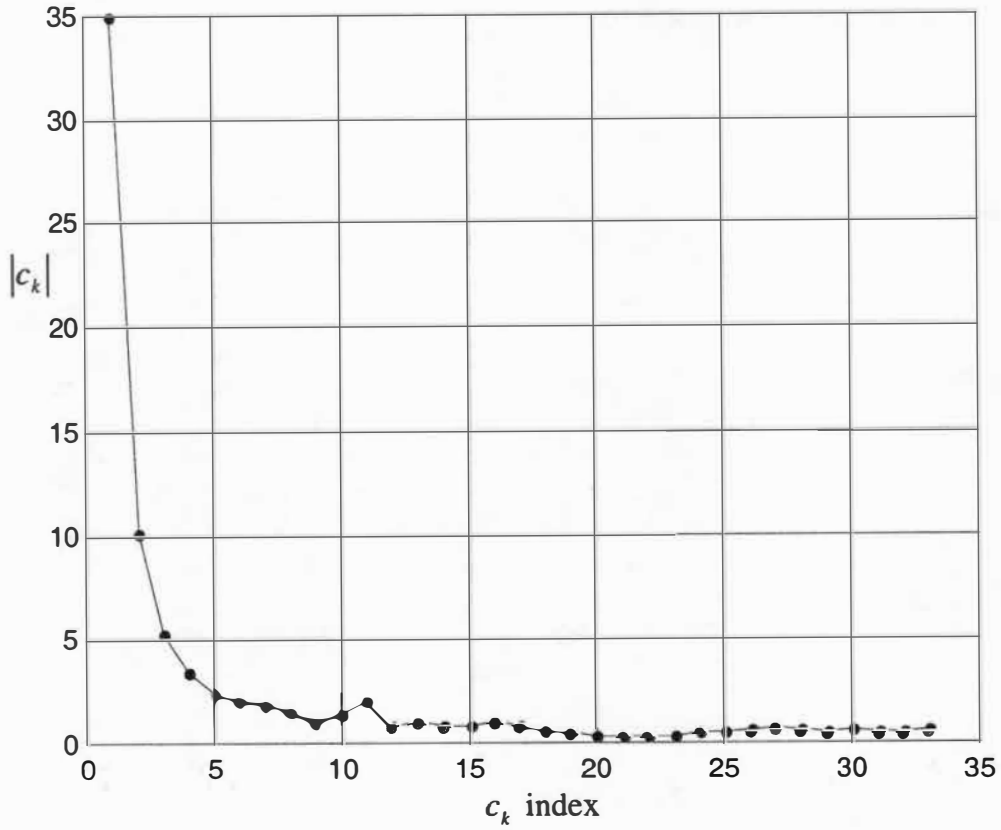

Figure 2.3.8 DFT of $f(n) = x_n + 0.1 \cdot r_n + 0.1 \cdot \sin(2\pi 10 x_n)$

A function composed of a sequence of straight lines, where the slopes of adjacent lines change by a factor of -1, characterize a triangle wave. Triangle waves, with slopes representing gradients, simulate in one dimension the gradients in twinning data. In Figure 2.3.9 f is the sum of a triangle wave with frequency 6 cycles in $[0,1]$ and a triangle wave with frequency 18 cycles in $[0,1]$. $|c_7|$ and $|c_{19}|$ identify $\lambda = 6$ and $\lambda = 18$, respectively, but several other non-zero coefficients are present although $f(n)$ does not have noise or linear trends added.

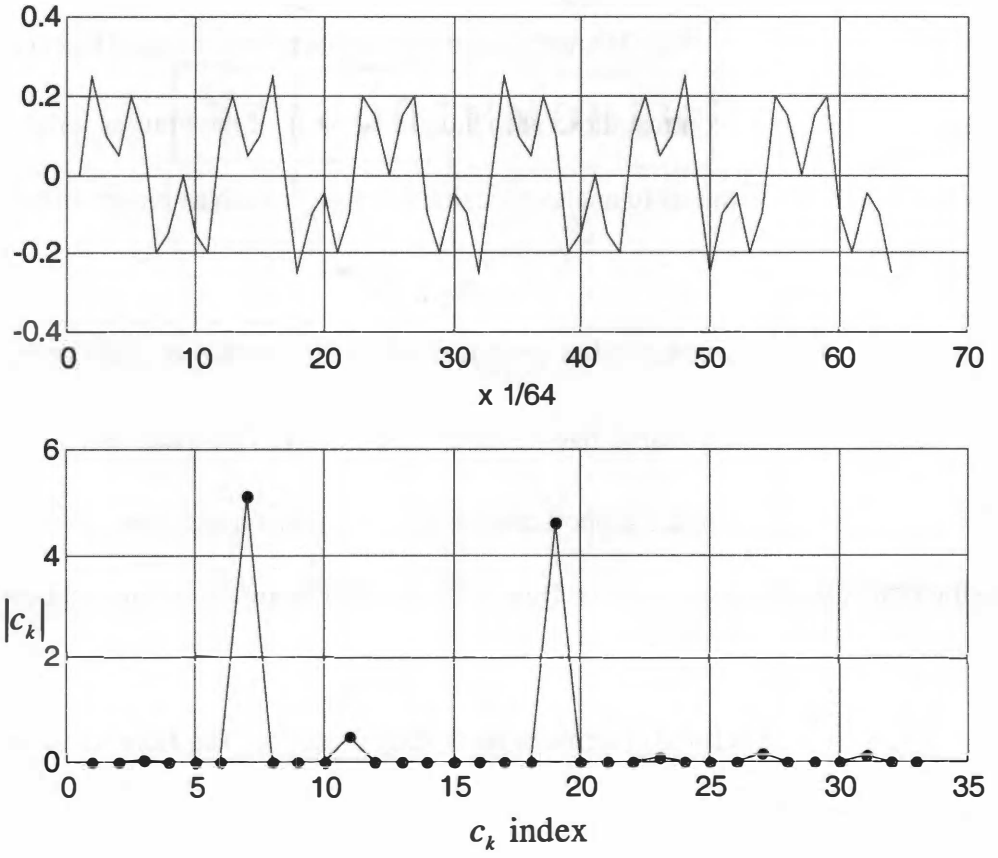

Figure 2.3.9 $f(n) = t(6x_n) + t(18x_n)$ (top), DFT of f (bottom).

Chapter 3

Wavelets

Section 3.1

Haar Wavelets

Wavelet transforms are better at capturing localized details of a function f than Fourier transforms since the basis functions of the wavelet transform have compact support and represent the function on a sequence of bounded subspaces. Wavelet basis functions have a variable support and translation term in the argument that allow us to represent any function $f \in L^2(\Re)$. We will construct the Haar wavelet and Daubechie-4 wavelet basis functions and contrast these with the global basis of the Fourier series. We will show how this will permit us to analyze localized behavior such as phase shifts and frequency changes.

The support of a function, written $spt(\phi)$, is the closure of the set $\{t|\phi(t) \neq 0\}$. If $spt(\phi)$ is compact then we say that ϕ has compact support. Both the Haar and Daubechie wavelets have compact support and allow us to define a basis on a closed and bounded interval. [Daubechie, p.3] The Haar wavelets will facilitate our description of Daubechie wavelets.

A wavelet basis is defined in terms of its scaling function. The Haar wavelet scaling function is defined to be

$$\phi(t) = \begin{cases} 1 & if\ 0 \leq t < 1 \\ 0 & otherwise \end{cases}. \qquad \textbf{(3.1.1)}$$

See Figure 3.3.1 for a graph of $\phi(t)$. Dilations and translations of ϕ yield changes in $spt(\phi)$. In the formula $\phi_{m,n}(t) = 2^{-m/2}(2^{-m}t - n)$ the scaling factor $2^{-m/2}$ and dilation term $2^{-m}t$ vary with m while the translation of $\phi(t)$ varies with n. The scaling factor is required to produce an orthonormal set of basis functions. We express the wavelet basis function ψ in terms of $\phi(t)$ by $\psi(t) = \phi(2t) - \phi(2t-1)$. We then define the basis functions $\{\psi_{m,n}\}$ by $\psi_{m,n}(t) = 2^{-m/2}\psi(2^{-m}t - n)$. In closed form, we have [Aboufadel, Schlicker p.6]

$$\psi_{0,0}(t) = \begin{cases} 1 & if\ 0 \le t < \frac{1}{2} \\ -1 & if\ \frac{1}{2} \le t < 1 \end{cases}. \qquad \textbf{(3.1.2)}$$

See Figure 3.1.2 for a graph of $\psi_{0,0}$. Graphs of $\psi_{1,0}$ and $\psi_{1,1}$ in Figure 3.1.3 show how $n = 0$ and $n = 1$ change the graph of $\psi_{1,n}$.

The set $\{\psi_{m,n}\}_{m,n}$ is an orthonormal set of functions and forms a basis for $L^2(\Re)$ with inner product defined to be $\langle f, g \rangle = \int f\bar{g}dx$. In other words the functions satisfy $\langle \psi_{m,n}, \psi_{m',n'} \rangle = \delta_{m,m'} \cdot \delta_{n,n'}$. Wavelet basis functions form a complete orthonormal basis for $L^2(\Re)$. [Daubechie p. 10]

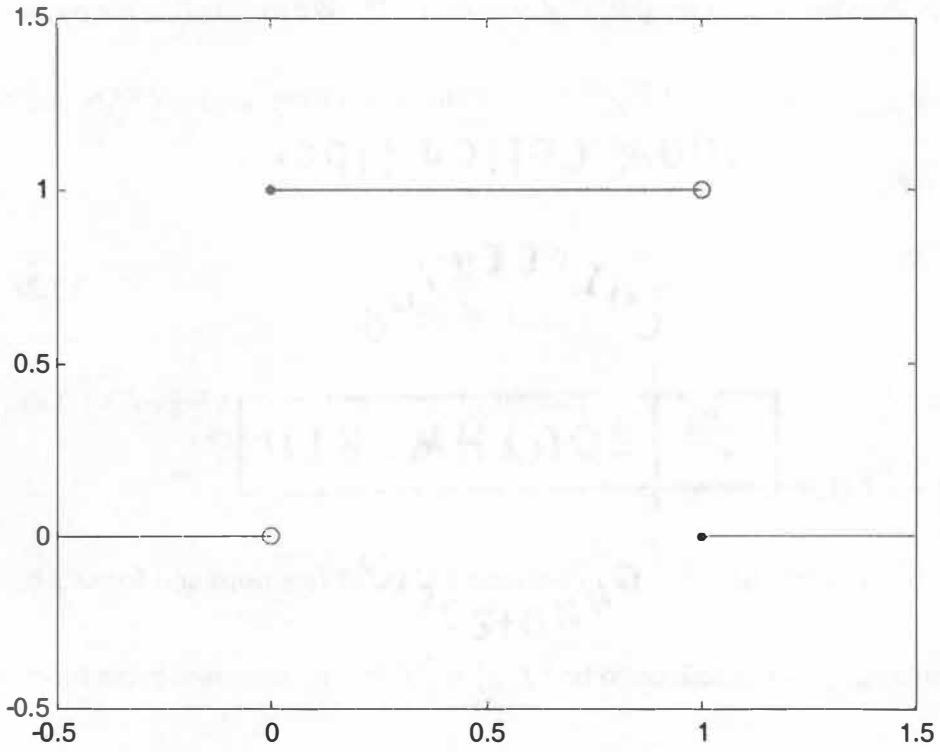

Figure 3.1.1 Haar scaling function $\phi(t)$.

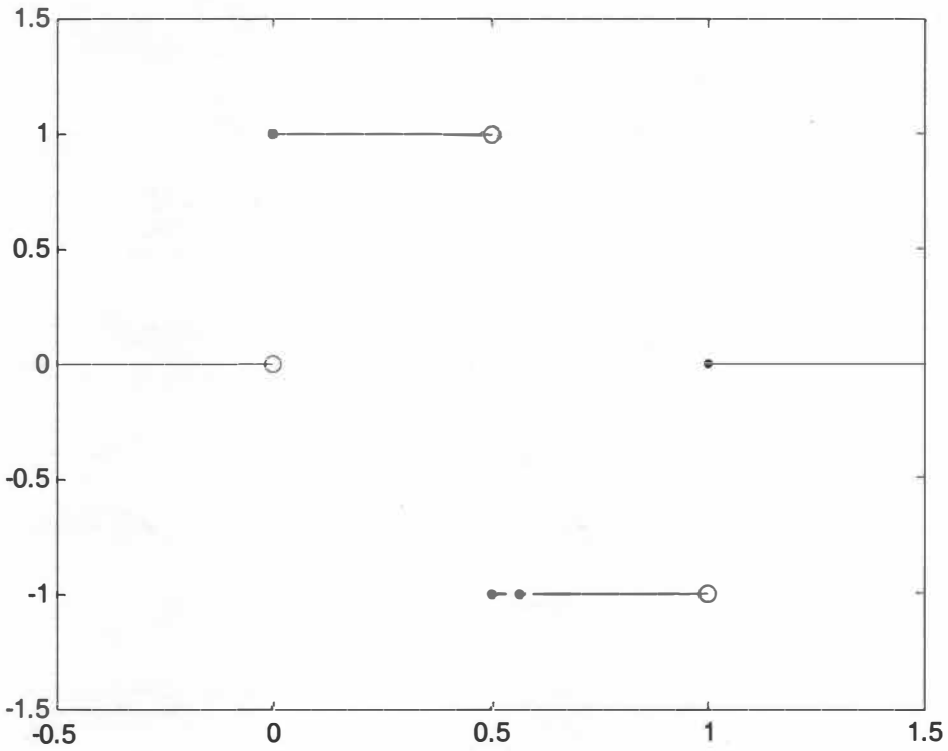

Figure 3.1.2 Haar basis function $\psi_{0,0}$.

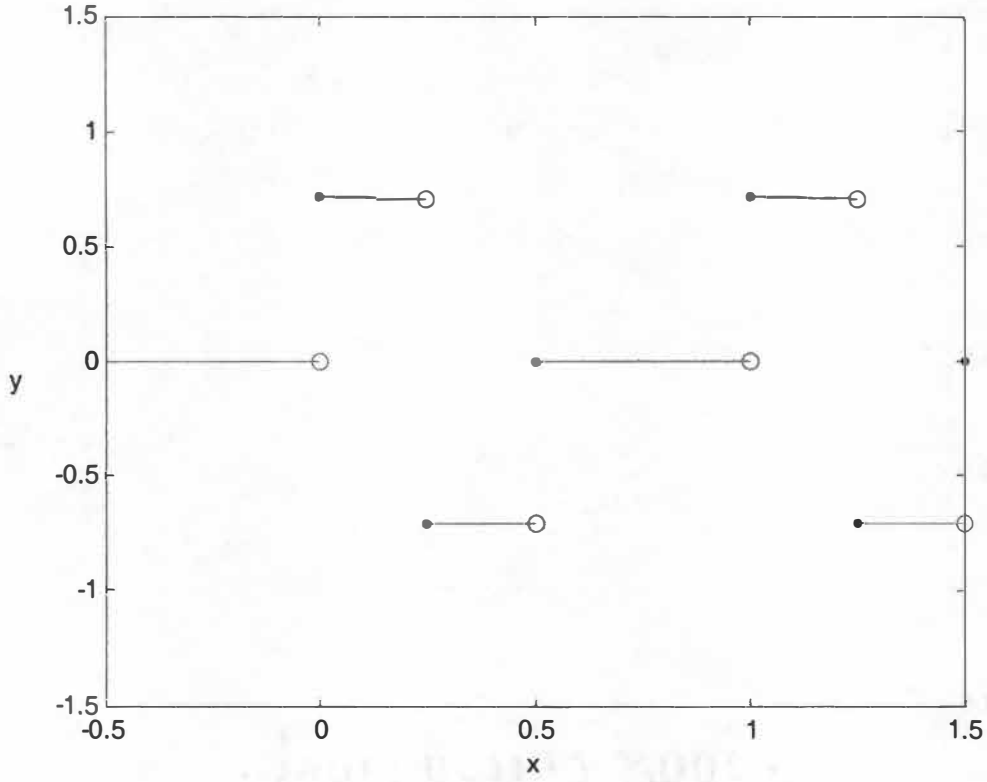

Figure 3.1.3 Haar basis functions $\psi_{1,0}$ and $\psi_{1,1}$ on intervals $[0,0.5]$ and $[1,1.5]$ respectively.

Section 3.2

Daubechie Wavelets

Haar wavelets are not suited for the problem of detecting characteristics of triangle waves but their formulation is similar to the Daubechie wavelets since both have: (1) basis functions with compact support and (2) dilation and translation terms in the arguments of $\phi_{m,n}$ and $\psi_{m,n}$ and (3) basis functions that are orthonormal with respect to dilation and translation. We chose the Daubechie-4, or D-4, wavelet since it is similar in form to a triangle wave and has similar properties to the Haar wavelet. Figure 3.2.1 has a graph of the D-4 scaling function $\phi(t)$ and Figure 3.2.2 has a graph of the D-4 detail ψ.

In general, Daubechie wavelet basis functions generate a set of subspaces that satisfy conditions for a multiresolution analysis. For $n \in Z$, let V_n be a subspace of $L^2(\Re)$. A multiresolution analysis is defined when a nested sequence of the subspaces V_n in $L^2(\Re)$, where $\cdots \subseteq V_{-1} \subseteq V_0 \subseteq V_1 \subseteq \cdots$, has a scaling function ϕ that spans each V_n and [Aboufadel, Schlicker p.54]

$$\bigcup_{n \in Z} V_n \text{ is dense in } L^2(\Re) \tag{3.2.1}$$

$$\bigcap_{n \in Z} V_n = \{0\} \tag{3.2.2}$$

$$f(t) \in V_n \text{ if and only if } f\left(2^{-n}t\right) \in V_0 \tag{3.2.3}$$

$$\{\phi(t-k)\}_{k \in Z} \text{ is an orthonormal basis for } V_0. \tag{3.2.4}$$

If these conditions are met then any $f \in L^2(\Re)$ can be approximated by a linear combination of basis functions generated by a wavelet scaling function. These conditions

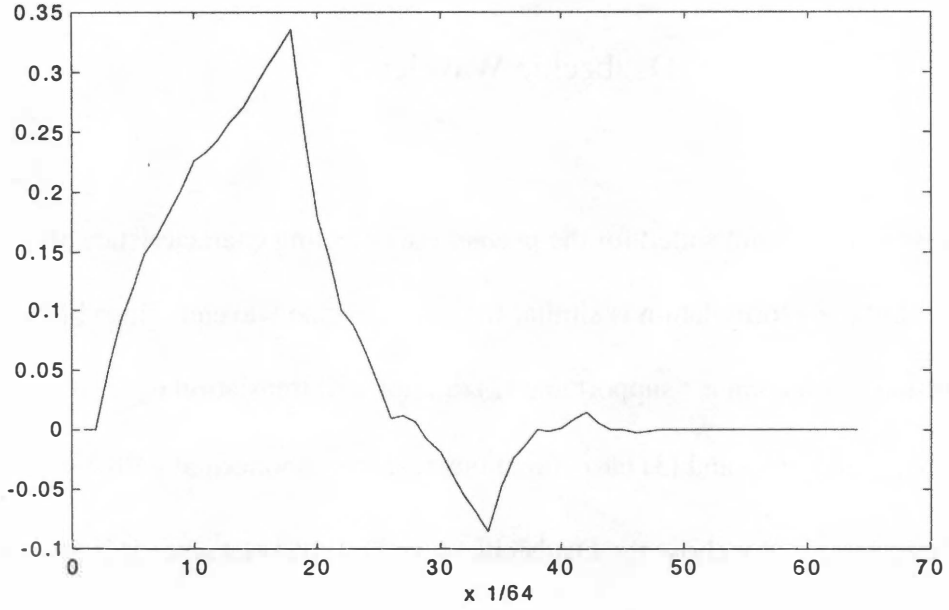

Figure 3.2.1 D-4 scaling function $\phi(t)$.

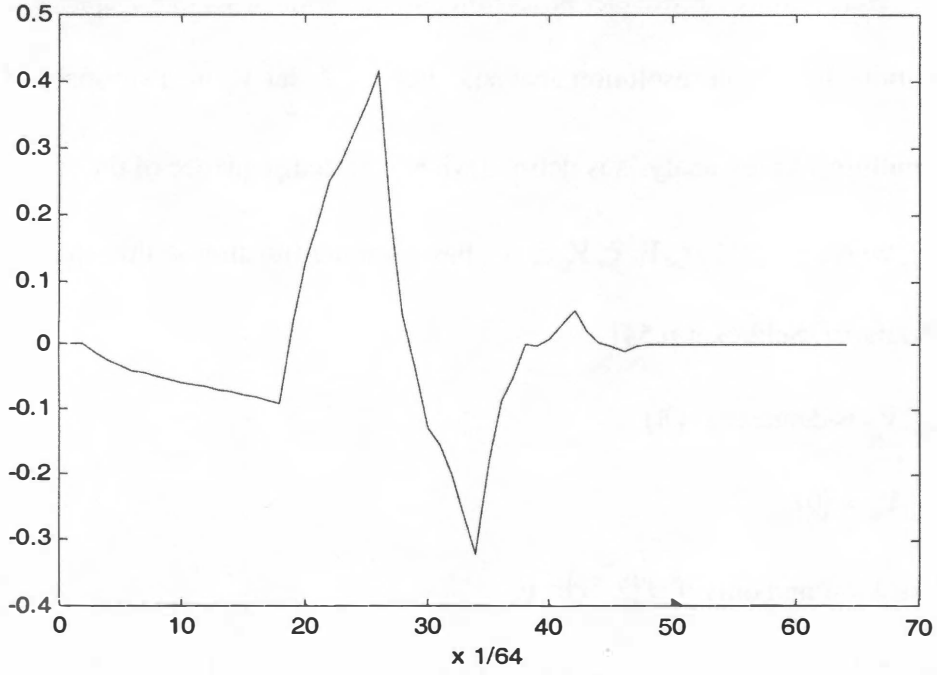

Figure 3.2.2 D-4 basis function ψ.

are satisfied by both the Haar functions, as defined in Section 3.1, and D-4 wavelets. However, it is easier to understand which subspaces are occupied by $\phi_{m,n}$ and $\psi_{m,n}$ when an explicit formula exists, such as in the Haar wavelets. D-4 wavelets have no explicit formula but must satisfy conditions (3.2.1) to (3.2.4). For example, Haar basis functions form a complete orthonormal basis in $L^2(\Re)$ so (3.2.1) is satisfied. Since the Haar wavelets $\phi_{m,n}$ and $\psi_{m,n}$ are such that $\phi_{m,n} \in V_m$ and $\psi_{m,n} \in V_m$, and $0 \in V_m \ \forall m$, the subspaces spanned by Haar wavelets satisfy condition (3.2.2). Haar wavelets have a dilation factor 2^{-m} that changes their support. Therefore they satisfy condition (3.2.3). In other words, $f(t) = \sum_n \alpha_n \phi_{m,n}(t)$ if and only if $f(2^{-m}t) = \sum_n \alpha_n \phi_{0,n}(t)$ where $\phi_{m,n} \in V_m$. Haar wavelets satisfy condition (3.2.4) since we have $\langle \phi_{m,n}, \phi_{m,n'} \rangle = \delta_{m,m'} \delta_{n,n'}$. For the D-4 wavelets, conditions (3.2.3) and (3.2.4) allow us to find coefficients c_k by relating $\phi(t-k) \in V_0$ to $\phi(2t-k) \in V_1$ in the equation $\phi(t) = \sum_k c_k \phi(2t-k)$. These coefficients allow us to write a discrete representation of the wavelet transform. [Aboufadel, Schlicker p.54] We obtain the numeric representation for the D-4 scaling function $\phi(t)$ with conditions [Aboufadel, Schlicker p.66, 67]

$$\phi(t) = 0 \text{ for } t \notin (0,3) \text{ and } \phi(t) \neq 0 \text{ a.e. for } t \in (0,3). \qquad \textbf{(3.2.5)}$$

$$\{\phi(t-k)\}_{k \in Z} \text{ is an orthonormal basis for } V_0. \qquad \textbf{(3.2.6)}$$

$$\int_{-\infty}^{\infty} \psi(t)dt = 0 \text{ and } \int_{-\infty}^{\infty} t\psi(t)dt = 0 \qquad \textbf{(3.2.7)}$$

If $\phi(t)=\sum_k c_k\phi(2t-k)$ where $\phi(t)\in V_0$ and $\phi(2t-k)\in V_1$, and $\langle\phi(t),\phi(2t-k)\rangle=0$ such that (3.2.1) is satisfied, then $c_k\neq 0$ for $k=0,\ldots,3$. Thus $\phi(t)=\sum_{k=0}^{3}c_k\phi(2t-k)$ and $\phi(t)$ has support on the interval $(0,3)$. Condition (3.2.2) requires that the scaling function $\phi(t)$ be orthonormal with respect to integer translates. Thus $\phi(t)$ has coefficients that satisfy $\left(\sum_k c_k^2\right)^{1/2}=\sqrt{2}$ and $\sum_k c_k c_{k-2m}=0$, $m\geq 1$. With $m=1$ in the last equation we have $c_0c_2+c_1c_3=0$. [Aboufadel, Schlicker p.67]

The wavelet basis function $\psi(t)$ is defined in terms of $\phi(t)$ with the sum

$$\psi(t)=\sum_{-\infty}^{\infty}(-1)^k c_{3-k}\phi(2t-k)$$

and since only the coefficients c_0,c_1,c_2, and c_3 are nonzero, the D-4 detail function is linear combination of $\phi(2t-k)\in V_1$. More explicitly,

$$\psi(t)=c_3\phi(2t)-c_2\phi(2t-1)+c_1\phi(2t-2)-c_0\phi(2t-3).$$

The moment conditions in (3.2.7) insure that $\phi(t)$ is smooth enough to approximate polynomials up to the first degree efficiently. In other words, by (3.3.3), $\langle f,\psi\rangle=0$ if f is linear. Two equations derived from the moment conditions for ψ, $-c_0+c_1-c_2+c_3=0$ and $-c_1+2c_2-3c_3=0$, give us the system of equations we use to find the coefficients c_0,c_1,c_2, *and* c_3. Thus we have the dilation equation

$$\phi(t)=\frac{1+\sqrt{3}}{4}\phi(2t)+\frac{3+\sqrt{3}}{4}\phi(2t-1)+\frac{3-\sqrt{3}}{4}\phi(2t-2)+\frac{1-\sqrt{3}}{4}\phi(2t-3)$$

that expresses $\phi(t)\in V_0$ in terms of $\phi(2t-k)\in V_1$. [Aboufadel, Schlicker p.66,67].

Section 3.3

Discrete Wavelet Transform

We denote functions in discrete form by $1xN$ arrays $f = \{f(n)\}$, where $f(n) = f(x_n)$ and $x_n = \frac{n-1}{N}$ for $n = 1,...,N$. The inner product of two functions is $\langle f, g \rangle = \sum_{n=1}^{N} f(n)g(n)$. Our subscript notation for $\phi_{m,n}$ and $\psi_{m,n}$ in the continuous case uses m for a subspace index. In the discrete case, m is associated with a level of wavelet analysis. First we have, in the continuous case, $\langle \phi_{m,n}, \psi_{m,n} \rangle = 0$ for all $m, n \in Z$. For fixed m, $\phi_{m,n} \in V_m$ and $\psi_{m,n} \in V_m^{\perp}$ for all n. We let W_m be the space spanned by $\{\psi_{m,n} : n \in Z\}$, where $W_m \subseteq V_m^{\perp}$. In the discrete case this leads to a finite number of subspaces. For arrays with length $N = 64$, the maximum index number is $m = 3$. The finest level of analysis is at $m = 3$ where we use the inner product formula $\langle f, \phi_{m,n} \rangle = \sum_{i=1}^{4} f(2n+i)\phi(i)$. To evaluate the inner product we express $\phi(i)$ and $\psi(i)$ in terms of their refinement coefficients as $\phi = [\frac{1+\sqrt{3}}{4}, \frac{3+\sqrt{3}}{4}, \frac{3-\sqrt{3}}{4}, \frac{1-\sqrt{3}}{4}]$ and $\psi = [\frac{1-\sqrt{3}}{4}, -\frac{3-\sqrt{3}}{4}, \frac{3+\sqrt{3}}{4}, -\frac{1+\sqrt{3}}{4}]$. We associate $\phi_{m,n} \in V_m$ and $\psi_{m,n} \in W_m$ for $m = \{0,1,2,3\}$ so we use eight subspaces or four levels of wavelet analysis of f. We use f only at the finest level and coefficients derived from ϕ at higher levels.

Data in the form of coefficients represent each level of analysis. The first level of detail coefficients is $c_{1,n}^{d} = \langle f, \psi_{3,n} \rangle$ and the first level of average coefficients is

$c^a_{1,n} = \langle f, \phi_{3,n} \rangle$ for $n = 1,...,33$. The second level of analysis uses the first level coefficients instead of f. We write $f_1^a(n) = c^a_{1,n}$ and use f_1^a to obtain second level coefficients given by $c^d_{2,n} = \langle f_1^a, \psi_{2,n} \rangle$ and $c^a_{2,n} = \langle f_1^a, \phi_{2,n} \rangle$ for $n = 1,...,18$. In general we have $f_m^a(n) = c^a_{m,n}$ so the third level of analysis gives us $c^d_{3,n} = \langle f_2^a, \phi_{1,n} \rangle$ and $c^a_{3,n} = \langle f_2^a, \phi_{1,n} \rangle$ for $n = 1,...,10$. Finally the fourth level of analysis gives us $c^d_{4,n} = \langle f_3^a, \psi_{0,n} \rangle$ and $c^a_{4,n} = \langle f_3^a, \phi_{0,n} \rangle$ for $n = 1,...,6$. We have just described Mallot's algorithm which is displayed in Figure 3.3.1.

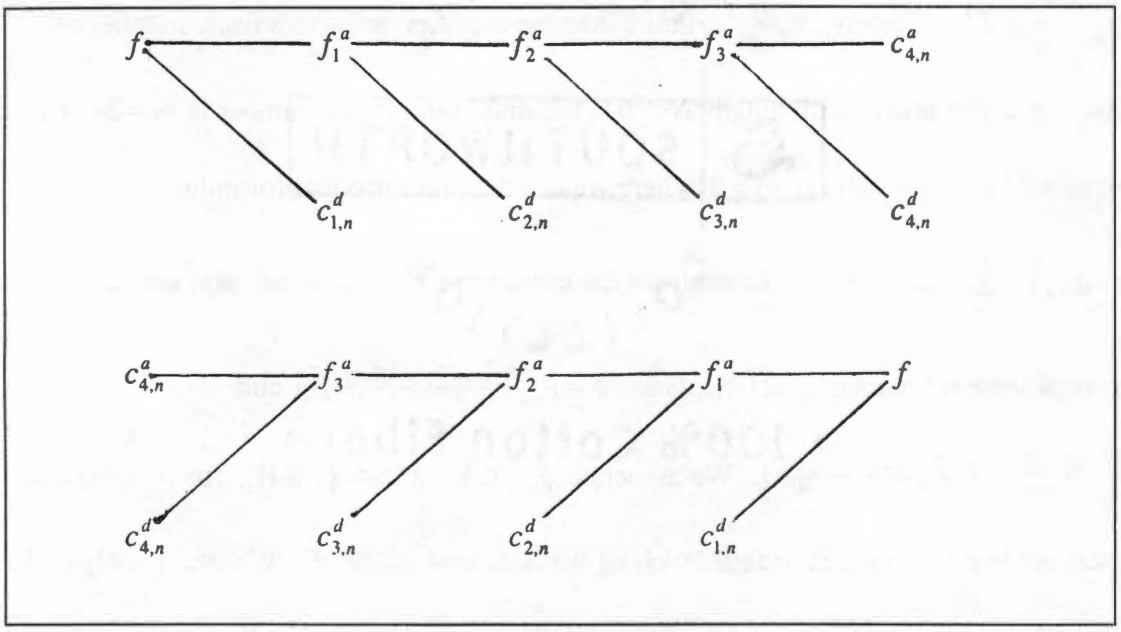

Figure 3.3.1 Mallot's Algorithm with DWT analysis (top) and reconstruction (bottom).

Section 3.4

Wavelet Transform Examples

As in the DFT examples, our examples in this section are $f(n) = f(x_n)$ where $x_n = \frac{n-1}{64}$ for $n = 1,...,64$. We have chosen examples to compare wavelet results with DFT results. There are four levels of analysis represented by DWT coefficients. In these examples the levels are graphed in order from the fourth level on the left to the first level on the right. The fourth level coefficients $c^a_{4,n}$ are represented by c_n for $n = 1,...,6$ while $c^d_{4,n}$ are represented by c_n for $n = 7,...,12$. The third level coefficients $c^d_{3,n}$ are represented by c_n for indices $n = 13,...,22$. The second level coefficients $c^d_{2,n}$ are represented by c_n for $n = 23,...,40$. Finally, the first level coefficients $c^d_{1,n}$ are represented by c_n and $n = 41,...,73$.

The first example given in Figure 3.4.1 is a graph of DWT coefficients for the delta function, where $f(n) = 0$ for $n = 1,...,15$ and $n = 17,...,64$ with $f(16) = 1$. The first level DWT coefficients are zero except for c_{48} and c_{49} and locate the position of the peak of the delta function. In contrast the DFT coefficients of a delta function (see Figure 2.3.2) are all non-zero.

In Figure 3.4.2 we have the DWT of a linear function. The moment conditions given by equations 3.2.7 guarantee that $c^d_{m,n} = 0$ except at the boundaries of f. To evaluate $c^d_{m,n}$ at the boundaries we used zero padding so $c^d_{m,n} \neq 0$ at boundaries between levels of coefficients. Most information is captured in c_n for $n = 1,...,6$.

DWT detail coefficients are not affected by linear trends as Figure 3.4.2 and moment conditions 3.2.7 suggest. Thus c_n in Figure 3.4.3 are similar to c_n of $f(n) = 0.1\sin(2\pi 10 x_n)$, except for boundary effects and c_n, $n = 1,...,6$.

In Figure 3.4.4 $f(n)$ is replaced by $g(n)$ for $n = 17,...,29$. The first level of DWT coefficients given by c_n, $n = 41,...,73$, clearly show a change in magnitude at $n = 51,...,54$. Compare this local change in DWT coefficients with DFT coefficients in Figure 2.3.4.

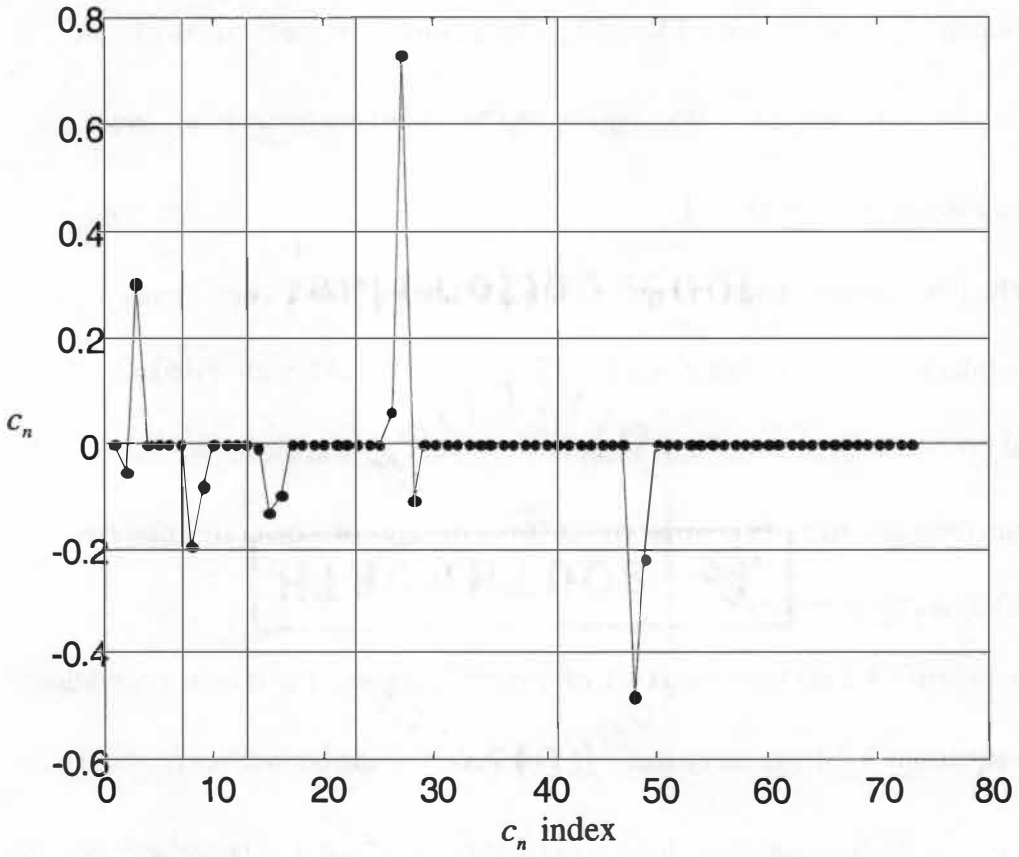

Figure 3.4.1 DWT wavelet coefficients c_n for a delta function f.

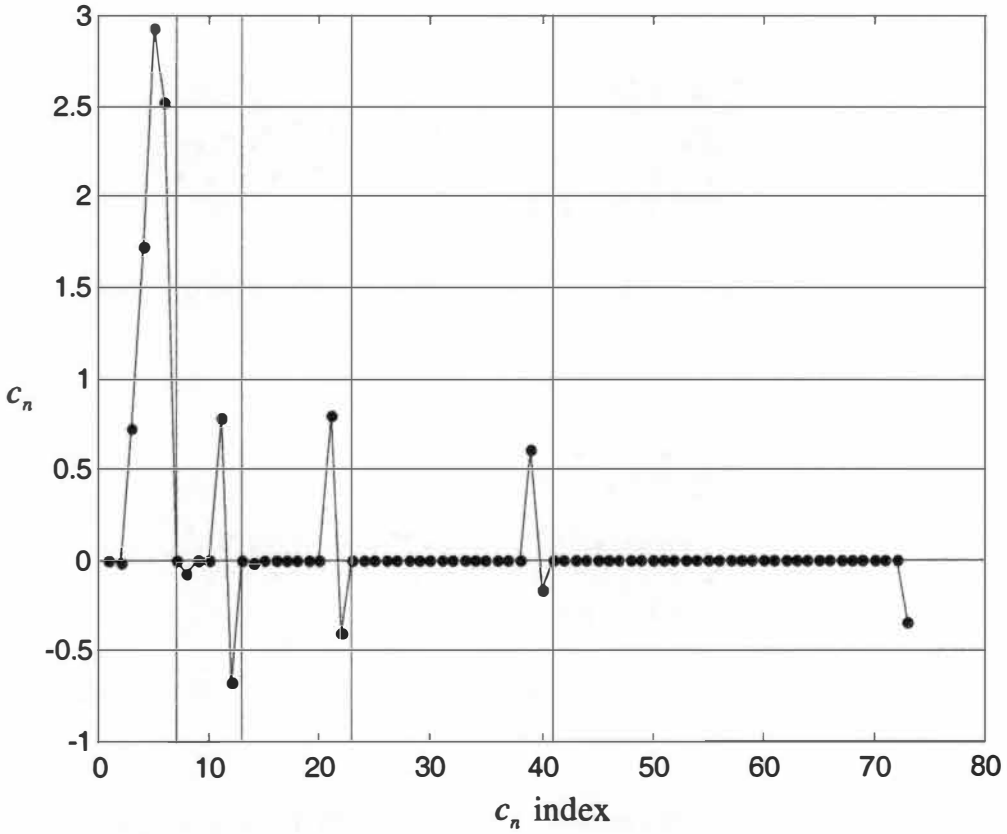

Figure 3.4.2 D-4 coefficients of $f(n) = x_n$

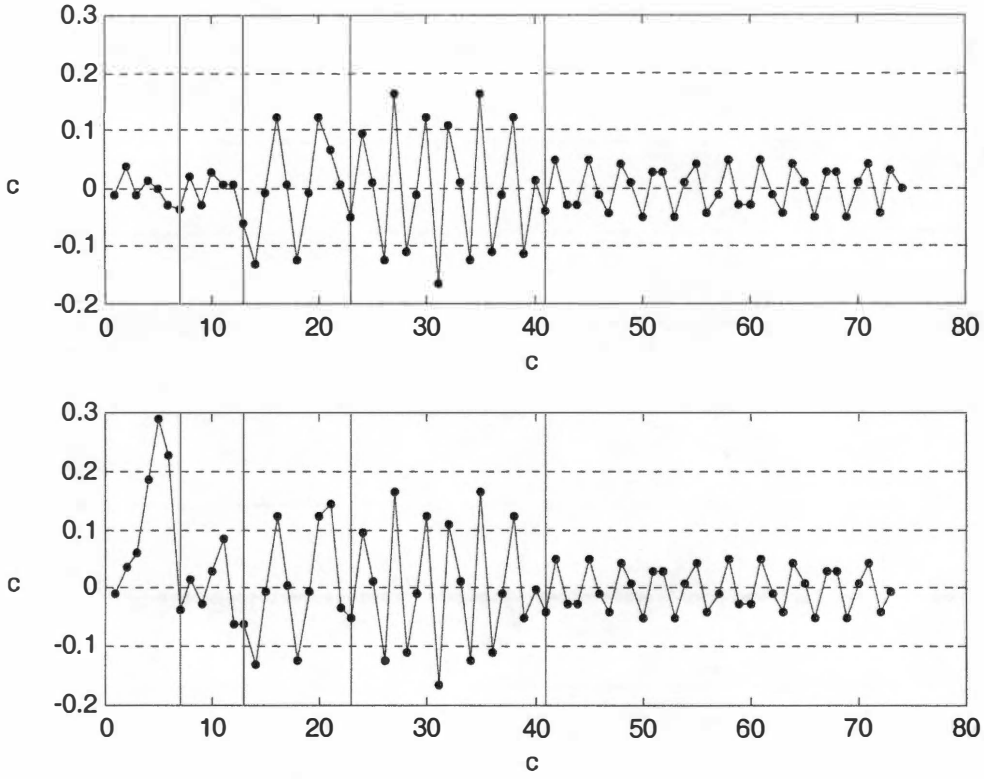

Figure 3.4.3 DWT coefficients of $f(n) = 0.1\sin(2\pi 10x_n)$ (top), and $f(n) = 0.1 \cdot x_n + 0.1\sin(2\pi 10x_n)$ (bottom).

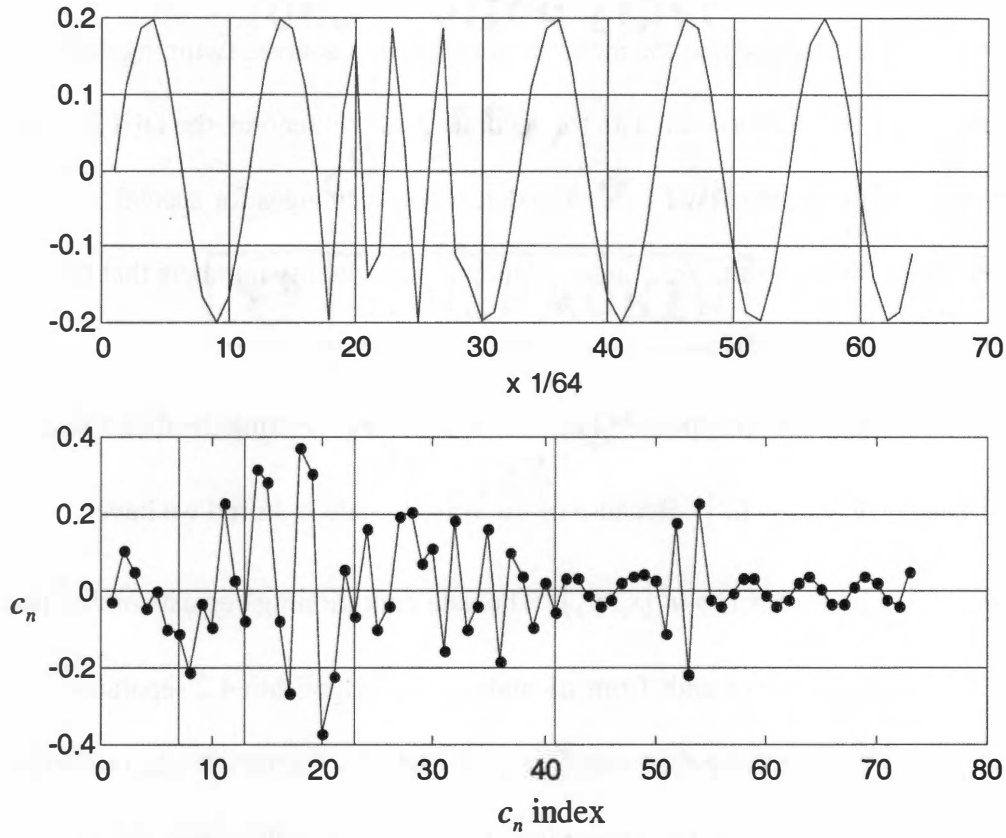

Figure 3.4.4 $f(n) = 0.2\sin(2\pi 6x_n)$ with $g(n) = 0.2\sin(2\pi 18x_n)$ at $n = 17,\ldots,29$ (top) DWT coefficients of $f(n)$ with $g(n)$ (bottom).

Chapter 4

Twinning Analysis

Section 4.1

Introduction

In this chapter we describe the methods developed to analyze twinning data. First, it is necessary to condition the data for accurate results. Second, the DFT is used to detect frequency. Finally, the DWT is used to detect phase changes for special frequencies. All of these results are combined to compute Quality numbers that measure the uniformity of twinning data.

The twinning data is composed of arrays $(u1, u2)$ representing the data values $\{u^k(x_i, y_j): 0 \le i, j \le N,\ k = 1,2\}$. Because of the way the data is stored we have $u1(j,i) = u^1(x_i, y_j)$ and $u2(j,i) = u^2(x_i, y_j)$. The data conditioning consists of two parts: Algorithm 4.1 removes linear trends from $u1$ and $u2$ and Algorithm 4.2 separates twinning information from noise then transfers the twinning information into one array.

We call the array that contains twinning information u with entries $u(i, j)$, $i, j = 1,...,N+1$. We analyze our data u row by row or column by column. We denote the i^{th} row by f_i^r, with $f_i^r(n) = u(i,n)$, $n = 1,...,N$ and the j^{th} column by f_i^c with $f_j^c(n) = u(n, j)$, $n = 1,...,N$. We eliminate the $N+1$ data value due to periodicity. We refer to the array f_i^r with f^r and the array f_i^r with f^c unless we need to specify a row or column. If no distinction between rows or columns is necessary we use f to refer to the array.

The main purpose of our Fourier analysis of twinning data is to identify the dominant trigonometric frequency λ_m and its associated Fourier coefficient c_m (See Section 4.3). To calculate c_m we use the conditioned data contained in u. [See section 4.2]. The motivation for our definition of the dominant trigonometric frequency of f is that we want to find the sine or cosine curve that best matches the shape of f. The frequency of that sine or cosine curve will be the dominant trigonometric frequency. In our case we define the best match to be the sine or cosine curve that best matches the slope of f pointwise.

Dominant Frequency Algorithm 4.3 (§ 4.3)

(1) Take the Fourier transform of f.
(2) Find the sine or cosine curve associated with the coefficients found in (1) to find the curve with the best pointwise slope match to f.

Algorithm 4.3 only gives integer values for λ_m but more precision was possible by using triangle waves to match the data. We found the frequency (λ), phase shift (β), and amplitude (α) of the triangle wave that was closest in norm to f in Algorithm 4.4 by using the Nelder-Mead Algorithm from Matlab to minimize $\|f - \alpha(\lambda(x_n - \beta))\|$, where λ_m and c_m are used to provide initial guesses for λ and β. [Hanselman]

One type of defect in data occurs when the uniformity of twinning data is interrupted by phase changes. The DWT indicates phase changes in f, for sufficiently high frequencies, when the sign of the first level coefficients from the DWT change. Thus, we can equate the number of sign changes in $c_{1,n}$ with the number of phase

changes in f. If $\lambda_m(f) \geq 24$, Algorithm 4.5 first finds the DWT of f then detects sign changes in $c_{1,n}$. We then find the number of times a sign change occurred in $c_{1,n}$.

We measure the uniformity of twinning data with the Quality Number. We collect statistics from Algorithms 4.4 and 4.5 applied to our data row by row and column by column, then we quantify variations in λ_m and phase changes in f.

Quality Number Algorithm 4.6 (§ 4.5)

(1) Find statistics of $\lambda_m(f)$ for rows and columns of u.
(2) Identify changes in $\lambda_m(f)$ for u.
(3) Count phase changes within each f.
(4) Use the Quality Number formula that is a function of (1), (2), and (3) to measure twinning data quality.

Section 4.2

Data Conditioning Algorithms

The data conditioning algorithms modify data to make it possible to accurately measure frequencies with Fourier transforms. Figures 2.3.6, 2.3.7 and 2.3.8 in Chapter 2 illustrate why it is necessary to remove linear trends before our analysis of data. The data in Figure 4.2.1 shows linear trends in the data arrays $u1$ and $u2$. Linear trends correspond to boundary conditions in the data given by the 2×2 matrix F in the equation $u(x, y) = F\begin{pmatrix} x \\ y \end{pmatrix}$ on the boundary.

Algorithm 4.1 [MODU (A.1)]

(1) Identify endpoints of linear trends and store in the 2×2 matrix F,

with $F_{1,1} = u1(1,m)$, $F_{1,2} = u1(m,1)$, $F_{2,1} = u2(1,m)$, and $F_{2,2} = u2(m,1)$.

(2) Remove linear trends from $u1$ and $u2$ to construct new matrices $v1$ and $v2$. Put matrices of x and y coordinates, X and Y of the reference lattice $L(e_i^0)$ in the equations $v1(i,j) = u1(i,j) - F_{1,1}X_i - F_{1,2}Y_j$ and $v2(i,j) = u2(i,j) - F_{2,1}X_i - F_{2,2}Y_j$. (See Figure 4.2.2)

Algorithm 4.2 [MIN2ND (A.2)] optimizes data for detecting frequencies by separating twinning data from noise. The twinning data is then transferred into one array from the $v1$ and $v2$ arrays. In Algorithm 4.2 we use the rotation matrix in the equation $\begin{pmatrix} w1 \\ w2 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} v1 \\ v2 \end{pmatrix}$ to find the angle θ that minimizes the norm of $w2$. We first find critical points of $\|v1\sin\theta + v2\cos\theta\|_F^2$ as follows. We equate the derivative of $\|v1\sin\theta + v2\cos\theta\|_F^2$ taken with respect to θ to 0, then solve for θ,

$$0 = \tfrac{d}{d\theta}\|v1\sin\theta + v2\cos\theta\|_F^2$$

$$= \tfrac{d}{d\theta}\left(\sum_{i,j}\left(v1_{ij}\sin\theta + v2_{ij}\cos\theta\right)^2\right)$$

$$= \sum_{i,j}\tfrac{d}{d\theta}\left(v1_{ij}\sin\theta + v2_{ij}\cos\theta\right)^2$$

$$= \sum_{i,j} 2\left(v1_{ij}\sin\theta + v2_{ij}\cos\theta\right)\left(v1_{ij}\cos\theta - v2_{ij}\sin\theta\right)$$

$$= 2\sum_{i,j}\left[(\sin\theta\cos\theta)\left((v1_{ij})^2 - (v2_{ij})^2\right) + (\cos^2\theta - \sin^2\theta)(v1_{ij}v2_{ij})\right]$$

$$= \sum_{i,j}\left[\left(\tfrac{1}{2}\sin 2\theta\right)\left((v1_{ij})^2 - (v2_{ij})^2\right) + (\cos 2\theta)(v1_{ij}v2_{ij})\right].$$

Thus

$$\tan 2\theta = \frac{-2\left(\sum_{i,j} v1_{ij}\, v2_{ij}\right)}{\sum_{i,j}\left(\left(v1_{ij}\right)^2 - \left(v2_{ij}\right)^2\right)}$$

or

$$\theta = \tfrac{1}{2}\arctan\frac{-2\sum_{i,j} v1_{ij}\, v2_{ij}}{\sum_{i,j}\left(\left(v1_{ij}\right)^2 - \left(v2_{ij}\right)^2\right)}.$$

From Algorithm 4.2 we obtain $w1$ and $w2$. We assume that the array with the largest norm contains twinning data. We take $u = \begin{cases} w1 \text{ if } \|w1\|_F > \|w2\|_F \\ w2 \text{ if } \|w2\|_F > \|w1\|_F \end{cases}$. For $w1$ and $w2$ used in the graph in Figure 4.2.3, $\|w1\|_F = 0.6911$ and $\|w2\|_F = 0.1420$ so $w1$ contains the twinning data. In Figure 4.2.4 the graphs of $f^r(n) = w1(32,n)$ and $g^r(n) = w2(32,n)$, $n = 1,\ldots,64$ illustrate how twinning data is different from the noisy part of the data. f^r is periodic and its amplitude is greater than g^r by an order of magnitude. It is obvious that g^r does not have a clearly defined period. From this point forward we will refer to the data that has linear trends and noise removed with u.

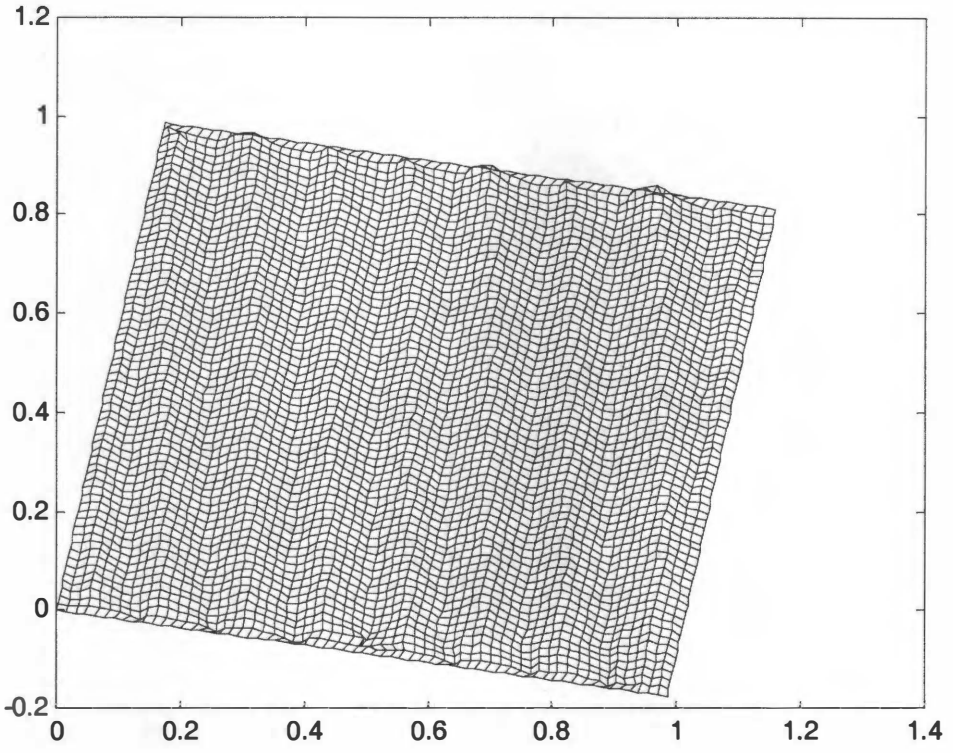

Figure 4.2.1 Plot of $u1$ verses $u2$.

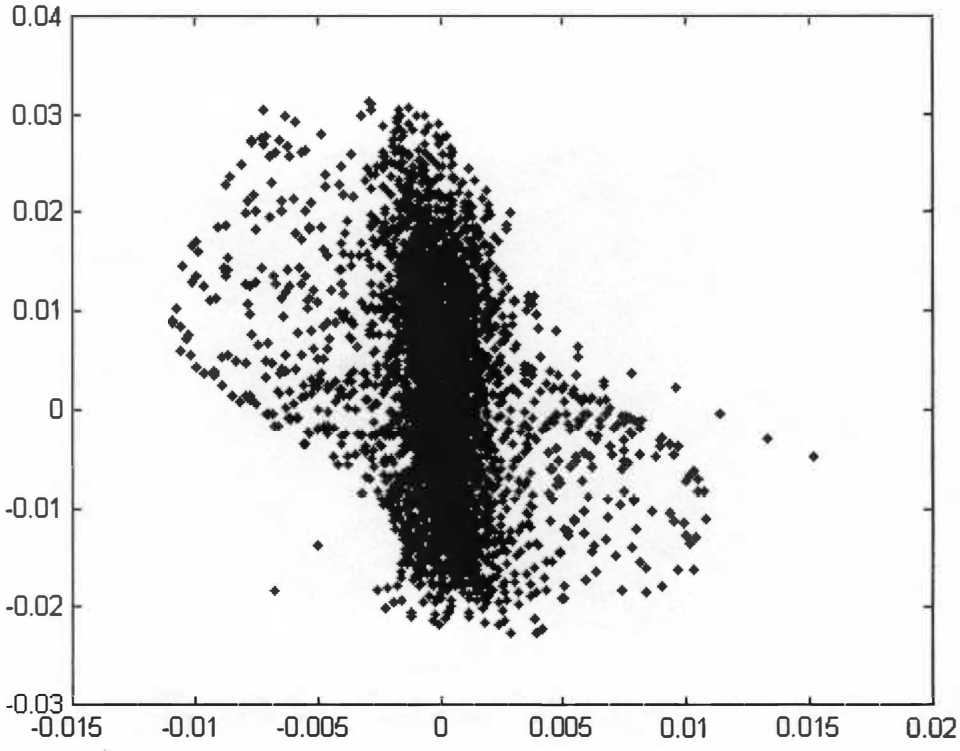

Figure 4.2.2 Plot of $v1$ verses $v2$.

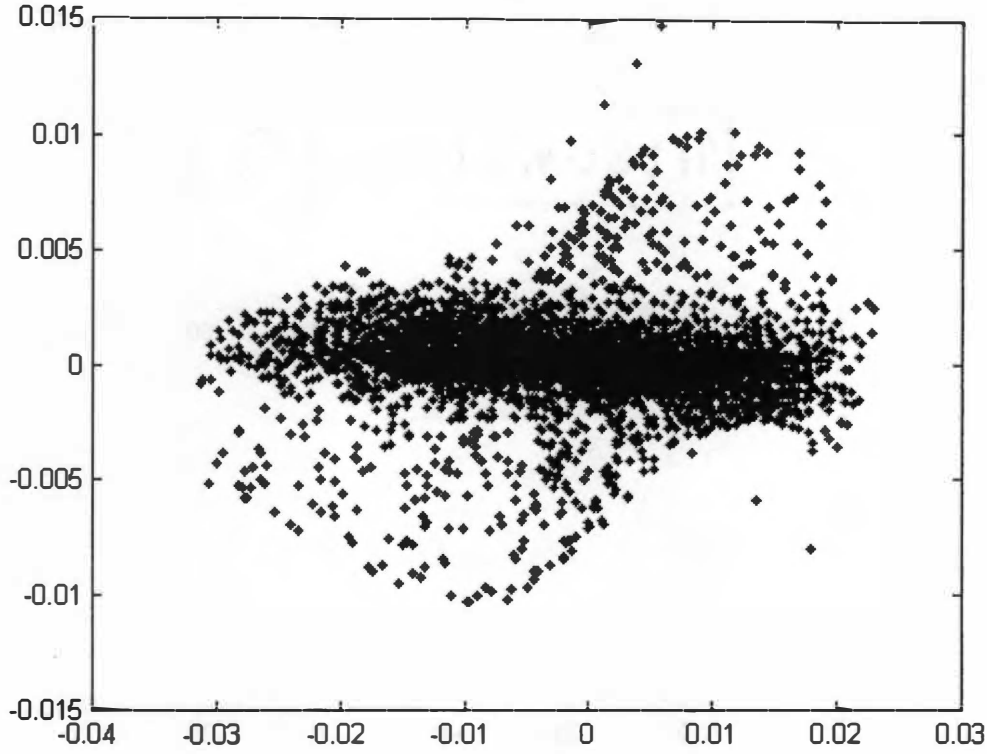

Figure 4.2.3 Plot of $w1$ verses $w2$.

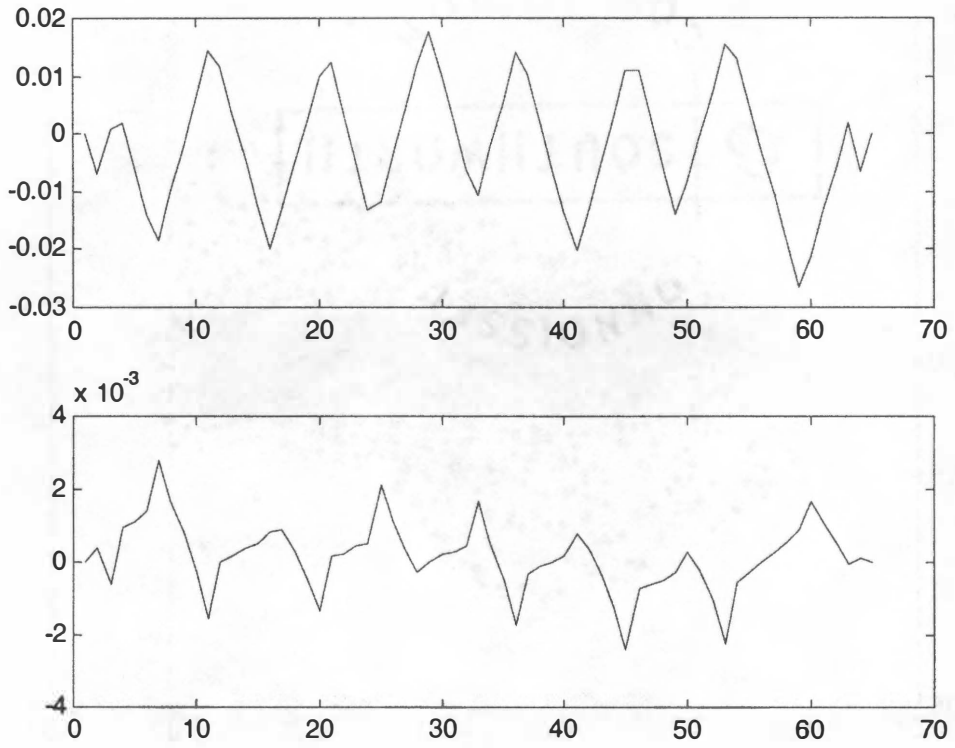

Figure 4.2.4 Graph of $f^{32}(n) = w1(32, n)$ at top and $g^{32}(n) = w2(32, n)$ at bottom.

Section 4.3

Frequency detection

Now we begin the process of finding dominant frequencies of u. We restrict our analysis to one-dimensional arrays of u. We use f^r and f^c to designate rows or columns of u. We take the DFT of f via

$$c_k(f) = \sum_{n=1}^{N} f(n) e^{-2\pi i (k-1)(n-1)/N} \text{ for } k = 1,\ldots,N. \qquad \textbf{(4.3.1)}$$

To match the frequency of f with the coefficients given by the Fourier transform, we developed an algorithm that compares the shape of f to the shape of the curve obtained from the inverse DFT of data containing the set of non-zero coefficients c_k and $\overline{c}_k$. Simply comparing the magnitudes of the Fourier coefficients of f does not correlate well with the dominant frequency λ_m. Example 2.3.7 shows that a coefficient related to a random term in f, in this case c_1, is larger in magnitude than the coefficient c_{11} when $\lambda_m = 10$ *cycles*. Additionally, our data is close in form to triangle waves and the DFT of triangle waves has non-zero coefficients that are not associated with the dominant frequency (See Figure 2.3.9.)

We use the inverse DFT transform defined in (2.2.1) to derive a function to compare to f. We construct an array of zeros with two non-zero entries. Recall that each coefficient in the DFT of a real function has a complex conjugate. In this case we have $c_k = \overline{c}_{N+2-k}$, where $\overline{c}_k$ is the complex conjugate of c_k. We build this array with $z_k(n) = 0$, $n = 1,2,\ldots,N$ and let $z_k(k) = c_k$ and $z_k(N+2-k) = \overline{c}_k$. Then define

$$\hat{f}_k(n) = \left(\frac{1}{N}\right)\sum_{j=1}^{N} z_k e^{2\pi i(j-1)(n-1)/N} \quad \text{for } n = 1,2,\ldots,N. \qquad \textbf{(4.3.2)}$$

We compare shapes by comparing the pointwise slope values between f and $\hat{f}_k$, so we use the discrete form of a derivative of a function: the divided difference.

Dominant Frequency Algorithm 4.3 [DFREQ (A.4)]

(1) Take the Fourier transform of f.
(2) Find $\hat{f}_k(n)$, $k = 2,\ldots,33$.
(3) Compare divided differences of f and $\hat{f}_k(n)$, $k = 2,\ldots,33$ to find best match.

To compare the shapes of f and $\hat{f}_k$ we take the sum of the product of the slopes of the arrays. We use divided differences to give the equivalent of slope between each pair of entries $f(n)$ and $f(n+1)$ in an array. Let f' be the divided difference of f so $f'(n) = f(n+1) - f(n)$. $\hat{f}_k'$ is defined in an analogous way. We use

$$\Gamma_k = \sum_{n=1}^{N} f'(n)\hat{f}_k'(n) \qquad \textbf{(4.3.3)}$$

to compare the shapes of $f(n)$ and $\hat{f}_k(n)$ for each k. Γ_k is large when $f'(n)$ and $\hat{f}_k'(n)$ have the same sign for all n, smaller or negative when $f'(n)$ and $\hat{f}_k'(n)$ have opposite signs for at least one n. Let

$$\Gamma_j = \max_k \Gamma_k. \qquad \textbf{(4.3.4)}$$

Then the dominant coefficient is $c_m = c_j$ and the dominant trigonometric frequency is

$$\lambda_m = j - 1 \qquad \textbf{(4.3.5)}$$

We use (4.3.1) to obtain a simpler form for (4.3.2). First we use the Inversion Property of the DFT to express f in terms of Fourier basis functions. Recall that we associate $\hat{f}_k$ with c_k in (4.3.2). To obtain this simpler form, we first shift the indices for DFT coefficients and DFT sums from $k = 1,...,N$ to $j = 0,...,N-1$. Let $\{c_j : j = 0,...,N-1\}$ be the set of DFT coefficients for f. Then

$f(n+1) = \frac{1}{N}\sum_{j=0}^{N-1} c_j e^{2\pi ij(n+1)/N}$ and $f(n) = \frac{1}{N}\sum_{j=0}^{N-1} c_j e^{2\pi ij(n)/N}$ for $n = 0,...,N-1$. The divided difference for f written in this form is

$$f'(n) = f(n+1) - f(n)$$

$$= \frac{1}{N}\sum_{j=0}^{N-1} c_j e^{2\pi ij(n+1)/N} - \frac{1}{N}\sum_{j=0}^{N-1} c_j e^{2\pi ijn/N}$$

$$= \frac{1}{N}\sum_{j=0}^{N-1} c_j \left(e^{2\pi ij(n+1)/N} - e^{2\pi ijn/N}\right)$$

$$= \frac{1}{N}\sum_{j=0}^{N-1} c_j e^{(2\pi ij/N)(n)}\left(e^{2\pi ij/N} - 1\right).$$

We use the formula (4.3.2) to find $\hat{f}_k(n)$. We shift the indices in (4.3.2) from $n = 1,...,N$ to $n = 0,...,N-1$ so there are two non-zero coefficients in the array z. We use z to compute $\hat{f}_k(n)$ are $z(k)$ and $z(N-k)$. Then we can rewrite (4.3.2) as

$$\hat{f}_k(n) = (\tfrac{1}{N})(c_k e^{2\pi ikn/N} + c_{N-k} e^{2\pi i(N-k)n/N}) \text{ for } n = 1,2,...,N-1$$

The divided difference $\hat{f}_k'$ becomes

$$\hat{f}_k'(n) = \hat{f}_k(n+1) - \hat{f}_k(n)$$

$$= \left(\tfrac{1}{N}\right)\left(c_k e^{2\pi ik(n+1)/N} + c_{N-k} e^{2\pi i(N-k)(n+1)/N} - c_k e^{2\pi ik(n)/N} - c_{N-k} e^{2\pi i(N-k)(n)/N}\right)$$

$$= \left(\tfrac{1}{N}\right)\left(c_k e^{2\pi ikn/N}\left(e^{2\pi ik/N} - 1\right) + c_{N-k} e^{2\pi i(N-k)n/N}\left(c_{N-k} e^{2\pi i(N-k)/N} - 1\right)\right).$$

We also shift the indices in (4.3.3) to $n = 0,\ldots,N-1$. Then we can express the sum of the product of divided differences $\hat{f}_k'$ and f' as

$$\Gamma_k = \sum_{n=0}^{N-1}\left[\left(\tfrac{1}{N}\right)\left(c_k e^{2\pi ikn/N}\left(e^{2\pi ik/N} - 1\right) + c_{N-k} e^{2\pi i(N-k)n/N}\left(e^{2\pi i(N-k)/N} - 1\right)\right)\cdot\right.$$

$$\left.\tfrac{1}{N}\sum_{j=0}^{N-1} c_j e^{(2\pi ij/N)n}\left(e^{2\pi ij/N} - 1\right)\right].$$

Since $e^{2\pi i(N-k)n/N} = e^{2\pi inN/N} e^{-2\pi ikn/N} = 1\cdot e^{-2\pi ikn/N}$ and

$e^{2\pi i(N-k)/N} = e^{2\pi iN/N} e^{-2\pi ik/N} = 1\cdot e^{-2\pi ik/N}$, we have

$$\tfrac{1}{N^2}\sum_{n=0}^{N-1}\left[c_k e^{2\pi ikn/N}\left(e^{2\pi ik/N} - 1\right) + c_{N-k} e^{-2\pi ikn/N}\left(e^{-2\pi ik/N} - 1\right)\right]\cdot\left[\sum_{j=0}^{N-1} c_j e^{(2\pi ij/N)n}\left(e^{2\pi ij/N} - 1\right)\right]$$

$$= \tfrac{1}{N^2}\sum_{j=0}^{N-1} c_j\left(e^{2\pi ij/N} - 1\right)\sum_{n=0}^{N-1}\left[c_k e^{2\pi ikn}\left(e^{2\pi ik/N} - 1\right) + c_{N-k} e^{-2\pi ikn/N}\left(e^{-2\pi ik/N} - 1\right)\right]\cdot e^{(2\pi ij/N)n}$$

$$= \tfrac{1}{N^2}\sum_{j=0}^{N-1} c_j\left(e^{2\pi ij/N} - 1\right)\sum_{n=0}^{N-1}\left[c_k e^{2\pi i(j+k)n}\left(e^{2\pi ik/N} - 1\right) + c_{N-k} e^{2\pi i(j-k)n/N}\left(e^{-2\pi ik/N} - 1\right)\right]$$

The inner sum is simplified to obtain

$$\tfrac{1}{N^2}\sum_{j=0}^{N-1} c_j\left(e^{2\pi ij/N} - 1\right)\cdot\left(c_k\left(e^{2\pi ik/N} - 1\right)\sum_{n=0}^{N-1} e^{2\pi i(j+k)n} + c_{N-k}\left(e^{-2\pi ik/N} - 1\right)\sum_{n=0}^{N-1} e^{2\pi i(j-k)n/N}\right).$$

Now we have $\sum_{n=0}^{N-1} e^{2\pi i(j+k)n/N} = \begin{cases} N & \text{if } j+k = N \\ 0 & \text{if } j+k \neq N \end{cases}$ and $\sum_{n=0}^{N-1} e^{2\pi i(j-k)n/N} = \begin{cases} N & \text{if } j = k \\ 0 & \text{if } j \neq k \end{cases}$.

Therefore

$$\tfrac{1}{N^2} c_{N-k}\left(e^{2\pi i(N-k)/N} - 1\right) c_k\left(e^{2\pi ik/N} - 1\right)N + \tfrac{1}{N^2} c_k\left(e^{2\pi ik/N} - 1\right) c_{N-k}\left(e^{-2\pi ik/N} - 1\right)N.$$

Finally, since $c_k = \overline{c}_{N-k}$ for the DFT of real functions, factoring and simplifying yield

$$\Gamma_k = \tfrac{2}{N}|c_k|^2\left(e^{2\pi ik/N} - 1\right)\left(e^{-2\pi ik/N} - 1\right)$$

$$= \tfrac{4}{N}|c_k|^2(1-\cos(\tfrac{2\pi k}{N})). \qquad \textbf{(4.3.5)}$$

We use (4.3.5) instead of (4.3.2) to compute Γ_k and thus $\lambda_m(f)$ in code [DFREQ (A.4)].

Dominant frequencies have only integer values due to properties of the DFT, which use only integer frequencies. Moreover, in Algorithm 4.3 we matched the slope change of f to the graphs of sine and cosine curves when our data that is closer in form to a triangle wave. Let t be a triangle wave. To improve the accuracy of measuring the frequency of the data, we minimize $\|f - \alpha t(\lambda(x-\beta))\|$ over the three variables α, β, and λ.

Minimization Algorithm 4.4 [SWTHCOND (A.5)]

(1) Find $\lambda_m(f)$ via Algorithm (4.3)
(2) Use $\lambda_m(f)$, α, and β as initial values for Nelder-Mead Simplex Algorithm [Matlab].
(3) Use Nelder-Mead Simplex Algorithm to minimize over variables in (2) over $\|f - \alpha t(\lambda(x-\beta))\|$.

We denote the dominant frequency by $\lambda_m(f)$ using this value of λ instead of the result from Algorithm 4.3 and equations (4.3.3), (4.3.4) and (4.3.5). Figure (4.3.1) has the graph of $f^{32}(n) = w1(32,n)$ with the graph of $\alpha t(\lambda_m x_n - \beta)$. We used Algorithms (4.3) and (4.4) to find $\alpha = 0.0144$, $\beta = 0.0066$ and $\lambda_m(f^{32}) = 7.5505$. The graph of the triangle wave t closely resembles the graph of f^{32} in amplitude and frequency. The phase angle match is visible where local maximums and minimums in f^{32} and t agree.

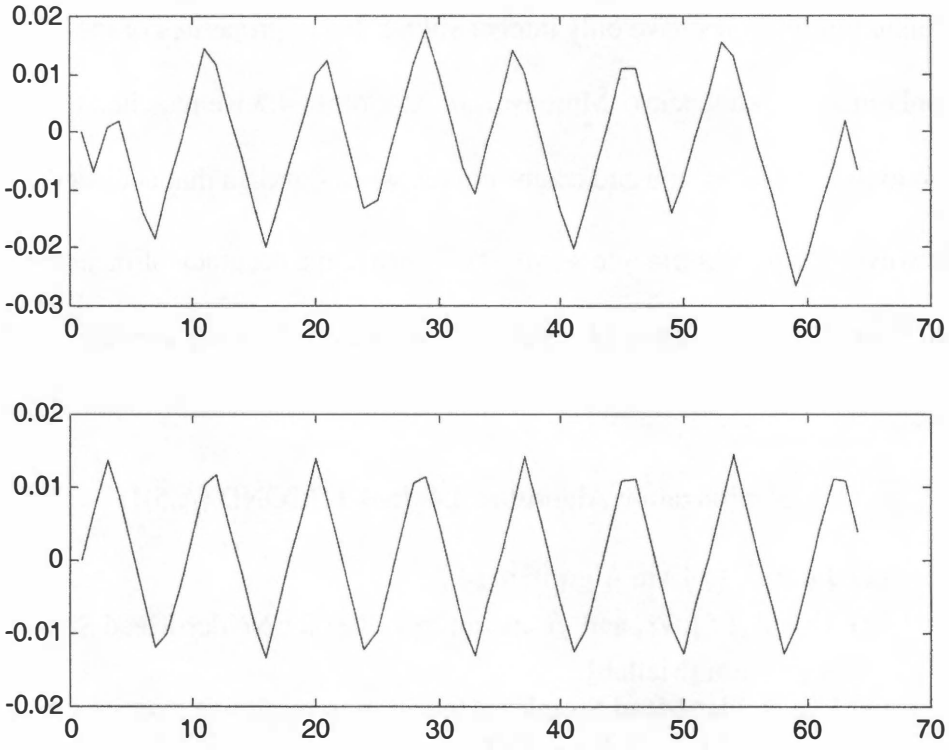

Figure 4.3.1 Graphs of $f^{32}(n) = w1(32,n)$ (top), and $t(n)$ (bottom).

Section 4.4

DWT Analysis

We use the Daubechie wavelets that satisfy two moment conditions. We refer to the discrete wavelet transform that uses Daubechie wavelets satisfying moment conditions as DWT. The scaling functions are $\phi_{m,n}$, and the wavelet basis functions are $\psi_{m,n}$. The index m refers to the level of analysis and the index n refers to translation. Recall that for the finest level of analysis we use the inner product formula

$$\langle f, \psi_{3,n} \rangle = \sum_{i=1}^{4} f(2n+i)\psi(i) \qquad (4.4.1)$$

to compute the coefficients $c_{1,n}^{d} = \langle f, \psi_{3,n} \rangle$, $n = 1,...,33$. We denote the sign of a coefficient with $sign(c)$. These coefficients are graphed in examples in Section 3.3 for DWT examples.

Wavelet coefficients give local information about f since the DWT coefficients are computed with (4.4.1). When the frequency is sufficiently high for the D-4 wavelets, i.e. when the period of a triangle wave occurs over four entries in f, the set of coefficients $\{c_{1,n}^{d} : n = 1,...,33\}$ are either all positive or all negative for a pure triangle wave. For an example, let $f = [a\,b\,c\,d]$ represent four entries of one wavelength where $a > b$, $b < c$, and $c > d$. The inner product, $\langle f, \psi \rangle$, is evaluated with refinement coefficients in $\psi = \left[\frac{1-\sqrt{3}}{4}, -\frac{3-\sqrt{3}}{4}, \frac{3+\sqrt{3}}{4}, -\frac{1+\sqrt{3}}{4}\right]$ so we have $\langle f, \psi \rangle = a\frac{1-\sqrt{3}}{4} - b\frac{3-\sqrt{3}}{4} + c\frac{3+\sqrt{3}}{4} - d\frac{1+\sqrt{3}}{4}$. Since DWT coefficients are computed with

(4.4.1), one coefficient contains localized information for four entries of f. Algorithm 4.5 [WVLTANAL (A.7)] takes the DWT of f if $\lambda_m \geq 25$. Then we find indices of $\{c_{1,n}^d\}$ such that $sign(c_{1,n}^d) = (-1)sign(c_{1,n-1}^d)$. We equate the number of sign changes in $\{c_{1,n}^d\}$ with the number of phase changes. We let $\rho(f)$ represent the number of sign changes for f.

In Figure 4.4.1, $f^r(n) = wl(30, n)$ is graphed at the top and the DWT coefficients $c_{1,n}^d$ of f^r are graphed at the bottom. $f^r(n)$changes phase at $c_{1,2n}^d$. There are a total of 7 phase changes so $\rho^r(f) = 7$.

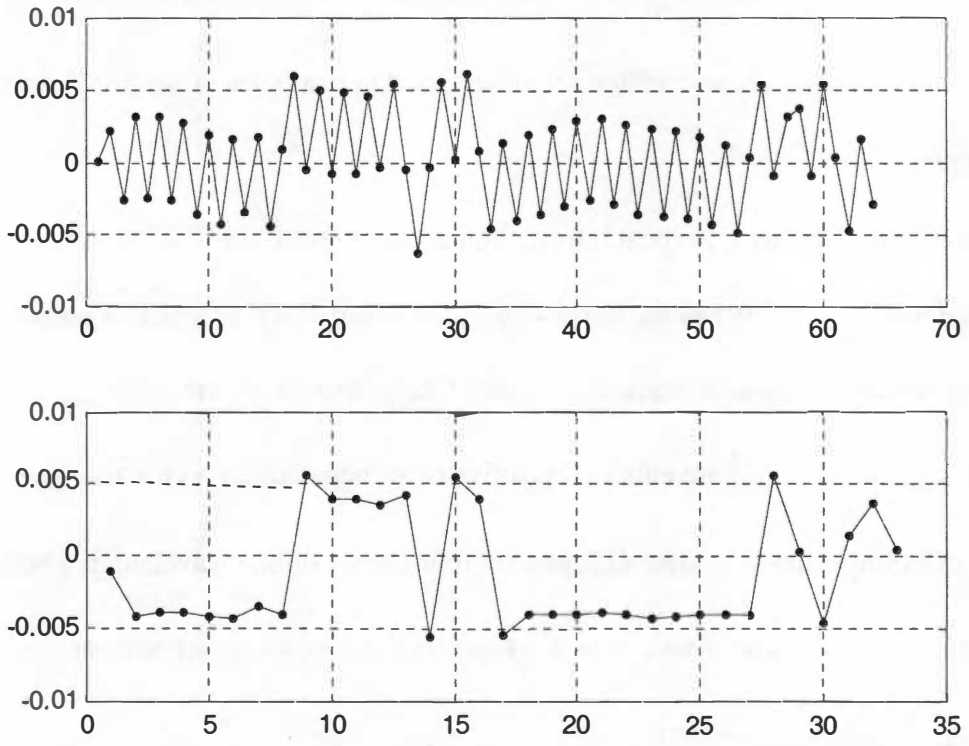

Figure 4.4.1 The graph of $f^r(n) = wl(30, n)$ (top), and $\{c_{1,n}^d\}$ (bottom).

Section 4.5

Quality Numbers

In this section statistics from DFT and DWT based algorithms (Algorithms 4.4 & 4.5), data size, and data norm are used to derive a quality number for u. We will discuss the DFT first.

We measure central tendency of the dominant frequencies from all rows and columns of twinning data with the mode and average. For the mode we let $\lambda_m^r(f^r)$, $r = 1,...,N$ represent dominant frequencies for the rows of u from Algorithm 4.4. We denote the mode frequency of the set of frequencies $\{\lambda_m^r(f^r): r = 1,...,N\}$ with Λ^r. We find the average of the set of frequencies $\{\hat{\lambda}_m^r\}$ with the formula $\hat{\lambda}_a^r = \frac{\sum \hat{\lambda}_m^r}{card\{\hat{\lambda}_m^r\}}$, the average frequency.

We use several measures of frequency dispersion to evaluate the uniformity of twinning data. One measure is to count how much of the data is near the mode Λ^r. We represent the subset of $\{\lambda_m^r\}$ such that $|\Lambda^r - \lambda_m^r| \le 0.5$ *cycles* with $\{\hat{\lambda}_m^r\}$ and represent the number of $\hat{\lambda}_m^r$ in the set of twinning data with $card\{\hat{\lambda}_m^r\}$. The second measure of dispersion is the number of times a sequence of $\{\hat{\lambda}_m^r\}$ is interrupted so we can count how many rows of data are defective with respect to Λ^r. To accomplish this we define δ^r to be the number of times $|\lambda_m^r - \lambda_m^{r-1}| > 1$ where only one of $\lambda_m^r = \hat{\lambda}_m^r$ or $\lambda_m^{r-1} = \hat{\lambda}_m^{r-1}$ is true.

Finally, we take the standard deviations of the set of frequencies $\{\lambda_m^r\}$ and represent that value with σ^r.

We also define a standard for the uniformity in the norm of f^r. The norm of f^r, $r = 1,...,N$, is given by $\|f^r\|_2$. For the quality number we define a norm for all f^r associated with $\hat{\lambda}_m^r$ with $\|f\|_{2,\hat{\lambda}_m} = \sum_{\hat{\lambda}_m} \|f^r\|_2$. This norm measures both the size of data using $\|f^r\|_2$, and the number of rows of data with frequency near $\hat{\lambda}_m^r$.

The DWT finds phase changes in f when the frequency is $\lambda \geq 25$ cycles. Each phase change is counted as a defect, so a large number of phase changes lessens the quality of u. We let ρ represent the number of phase changes in f and let ρ^r be the average number of phase changes in the set $\{f^r\}$. All statistical quantities are defined in an analogous way for $\{f^c\}$ and will be marked with a c.

The formula for twinning quality was developed to reward uniform data with a high numeric score. We use two terms, one for rows and one for columns. For the rows we divide $\|f\|_{2,\hat{\lambda}_m}$ by the sum of two terms, $N + \frac{\sigma^r + \delta^r + \rho^r}{\hat{\lambda}_a^r}$. The norm $\|f\|_{2,\hat{\lambda}_m}$ is a measure of both the number of rows associated with Λ^r and the size of f. If twinning data contains few rows with frequencies near Λ^r or if the twinning data has a small norm, the score is lower since the size of the divisor N remains constant. To assign lower scores to data with data with dispersed frequencies, we add the variable term $\frac{\sigma^r + \delta^r + \rho^r}{\hat{\lambda}_a^r}$ to N. The numerator $\sigma^r + \delta^r + \rho^r$ contains three measures of data

lacking uniformity: the standard deviation of the row frequencies, the number of interruptions in a sequence of $\{\hat{\lambda}^r_m\}$, and the average number of phase shifts for all of the rows of twinning data. The score is lower for data when these dispersive measures are high. The denominator $\hat{\lambda}^r_a$ scales the dispersive measures σ^r, δ^r, and ρ^r by the frequency since we expect larger scores for σ^r and δ^r for high frequency data. In addition ρ^r is not evaluated for data with a frequency $\lambda < 25$. Again, we treat the term for columns the same way.

The Quality Number formula [QNUMS (A.11)] used to measure twinning data uniformity is

$$Q = \frac{\|f^r\|_{2,k_m}}{N + \dfrac{\sigma^r + \delta^r + \rho^r}{\hat{\lambda}^r_a}} + \frac{\|f^c\|_{2,k_n}}{N + \dfrac{\sigma^c + \delta^c + \rho^c}{\hat{\lambda}^c_a}}.$$

Using test data generated to be as uniform as possible, we scaled Q by the maximum value of Q found from all these sets of twinning data test data. Within this set of data, the highest possible score is 100, but it is possible to obtain a higher score in other sets of data. From this point forward we will refer to Quality numbers with Q-number.

Section 4.6

Results

We developed our algorithms to detect uniformity in the banded structures in the graphs of twinning data so we expect high Q-numbers when graphs of twinning uniform band patterns. However, twinning data can have characteristics that have a negative effect on the quality number. We found that Q-numbers tended to be low for high frequency data or data with bands oriented in more than one direction.

The Q-number equation was developed by collecting statistics on test data that was generated to be as uniform as possible. Sets of test data are represented in Figures 4.6.1-4. Sets of data represented in Figures 4.6.6-10 were generated with no initial preference for uniform structure. Data represented in Figures 4.6.9 and 4.6.10 have bands with more than one orientation. All sets of data used $N = 64$ in the Q-number formula.

The data with the highest Q-number is graphed in Figure 4.6.1 and it is clear that the twinning bands are uniform. Since all Q-numbers are scaled by the value found for this set, we have $Q = 100$. There were several reasons for the high score. The number of rows and columns of data with frequencies near Λ^r and Λ^c are $card\{\hat{\lambda}_m^r\} = 63$ and $card\{\hat{\lambda}_m^c\} = 63$. The average norms of data with frequencies near the mode frequency are $\|f\|_{2,\hat{\lambda}_m^r} = 0.0810$ and $\|f\|_{2,\hat{\lambda}_m^c} = 0.0814$. The highest value for this norm in the test data was 0.1217 and the lowest was 0.0308. The average standard deviation of the

frequencies were $\sigma^r = 1.9218$ and $\sigma^c = 1.4795$. The range of average standard deviations was from 1.4690 to 14.7672.

In Figure 4.6.2 we obtained $Q = 95$ for similar reasons; $card\{\hat{\lambda}_m^r\} = 61$ and $card\{\hat{\lambda}_m^c\} = 63$, $\|f\|_{2,\hat{\lambda}_m^r} = 0.0963$ and $\|f\|_{2,\hat{\lambda}_m^c} = 0.0940$, and $\sigma^r = 4.0626$ and $\sigma^c = 3.4157$. The bands are not as uniform as in Figure 4.5.1 and this was detected by σ^r and σ^c.

In Figure 4.6.3 we obtained $Q = 39$. The bands are not nearly as uniform as the bands in Figure 4.6.1 and Figure 4.5.2. Statistics that lower the Q-number are $card\{\hat{\lambda}_m^r\} = 31$, $card\{\hat{\lambda}_m^c\} = 24$, $\sigma^r = 9.8763$, and $\sigma^c = 10.2557$.

Figure 4.6.4 has a low Q-number despite what appears to be the most uniform bands of all the data we analyzed. The statistic that is responsible for $Q = 64$ was $\sigma^r = 14.7672$, the highest σ^r in all of the test data, and $card\{\hat{\lambda}_m^r\} = 37$ so that we have $\|f\|_{2,\hat{\lambda}_m^r} = 0.0335$, a low number. We also obtained $\|f\|_{2,\hat{\lambda}_m^c} = 0.0340$. The underlying reason for this was faulty frequency detection since $\Lambda^r = 2$ although we should consistently have $\Lambda^r = 32$. Figure 4.6.5 is the graph of a typical row of the data. The arch in the data affected dominant trigonometric frequency detection in Algorithm 4.3.

Data generated without regard for band uniformity is graphed in Figure 4.6.6. This set of data has $Q = 99$. Although $card\{\hat{\lambda}_m^r\} = 37$, $card\{\hat{\lambda}_m^c\} = 53$, the norms $\|f\|_{2,\hat{\lambda}_m^r} = 0.1672$ and $\|f\|_{2,\hat{\lambda}_m^c} = 0.1758$ were the highest results we obtained in all of our data.

Figure 4.6.7 is another example of high frequency data with a low score and there is clearly less uniformity in the bands than in Figure 4.6.4 where $Q = 64$. In this case $Q = 56$ with $card\{\hat{\lambda}_m^r\} = 37$ and $card\{\hat{\lambda}_m^c\} = 53$. The norms $\|f\|_{2,\hat{\lambda}_m^r} = 0.0395$ and $\|f\|_{2,\hat{\lambda}_m^c} = 0.0407$ were among the lowest. The number of phase shifts $\rho^r = 8.2031$, also lowered the score.

Figure 4.6.8 has data with bands that are clearly not uniform in width. Some bands have a slightly different orientation as well. In this case $Q = 31$, which is low because $\|f\|_{2,\hat{\lambda}_m^r} = 0.0434$ and $\|f\|_{2,\hat{\lambda}_m^c} = 0.0295$ are among the lowest in our data.

Figure 4.6.9 is a graph of data with bands having more than one orientation. In this case $Q = 52$. Although $\|f\|_{2,\hat{\lambda}_m^r} = 0.1492$ and $\|f\|_{2,\hat{\lambda}_m^c} = 0.1147$ are large, we had $card\{\hat{\lambda}_m^r\} = 37$ and $card\{\hat{\lambda}_m^c\} = 53$, which are low numbers.

Figure 4.6.10 is an example of twinning data with bands oriented in more than on direction but has a high score for this type of data, $Q = 82$. Statistics that helped this Q-number are $card\{\hat{\lambda}_m^r\} = 51$ and $card\{\hat{\lambda}_m^c\} = 57$, and $\|f\|_{2,\hat{\lambda}_m^r} = 0.1492$ and $\|f\|_{2,\hat{\lambda}_m^c} = 0.1147$.

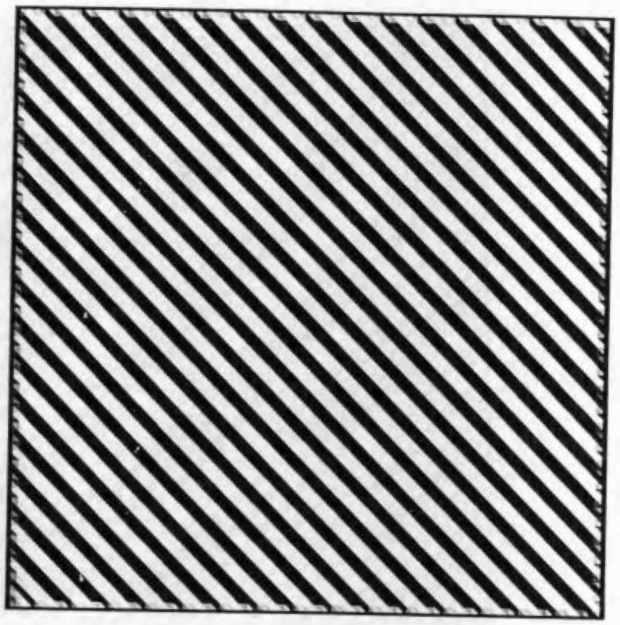

Figure 4.6.1 Twinning data with Q = 100.

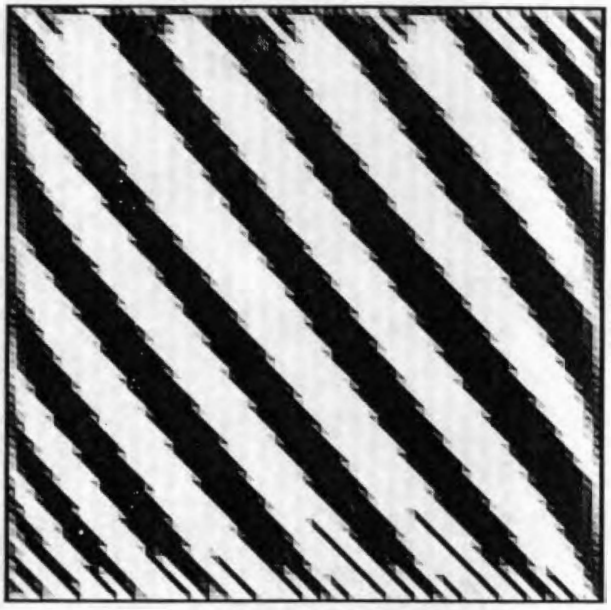

Figure 4.6.2 Twinning data with Q = 95.

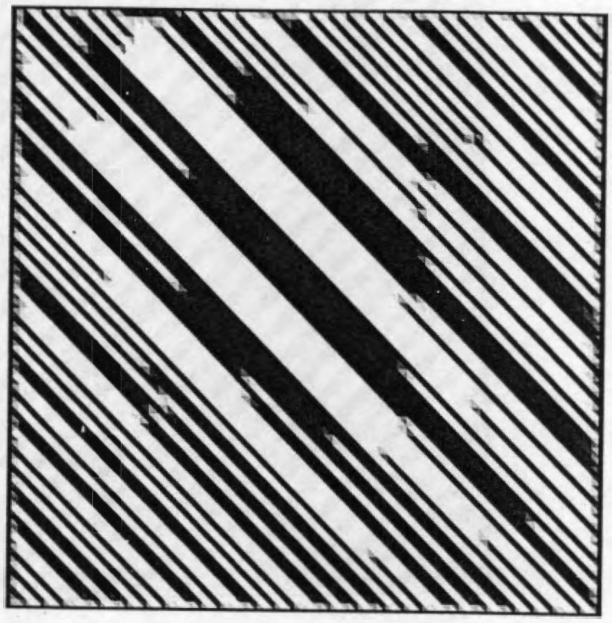

Figure 4.6.3 Twinning data with Q = 39.

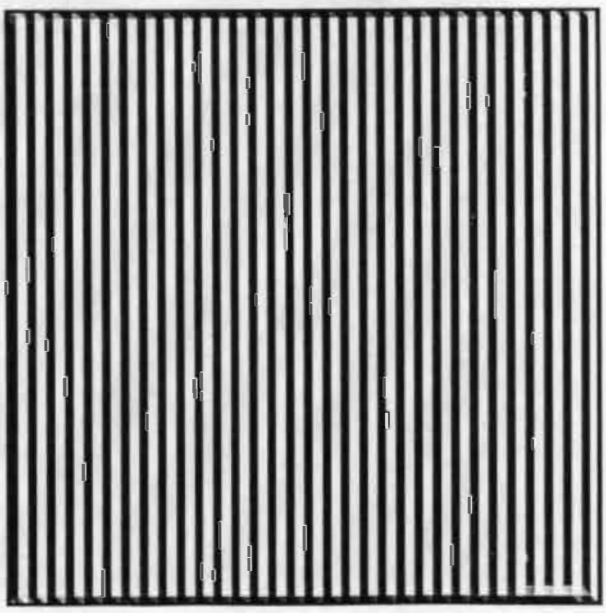

Figure 4.6.4 Twinning data with Q = 64.

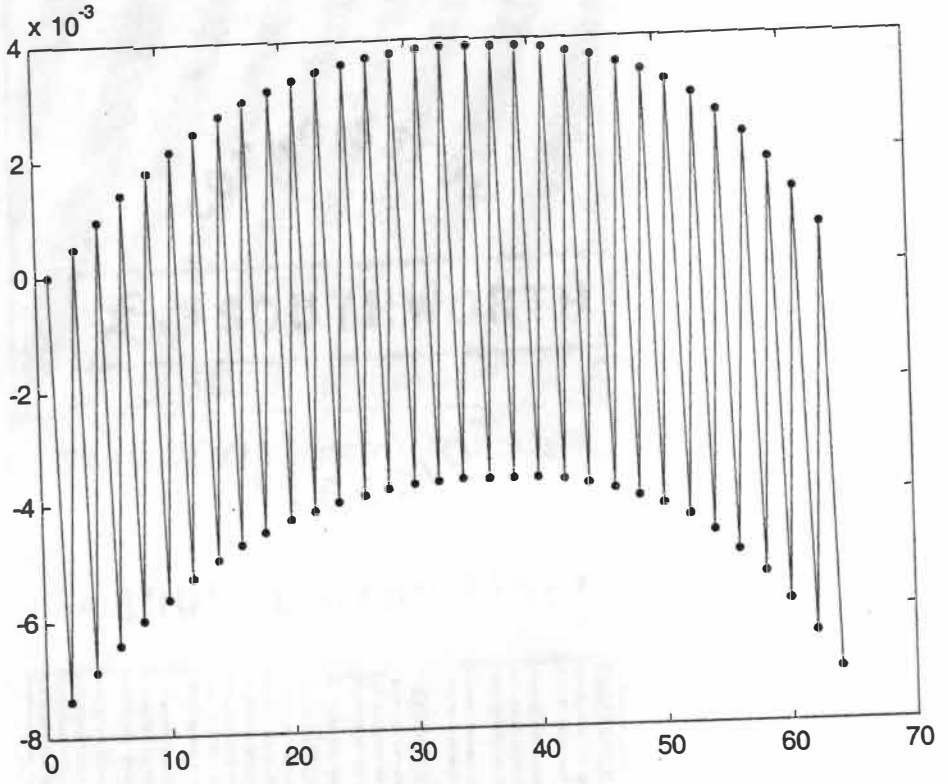

Figure 4.6.5 Typical row data from Figure 4.6.4.

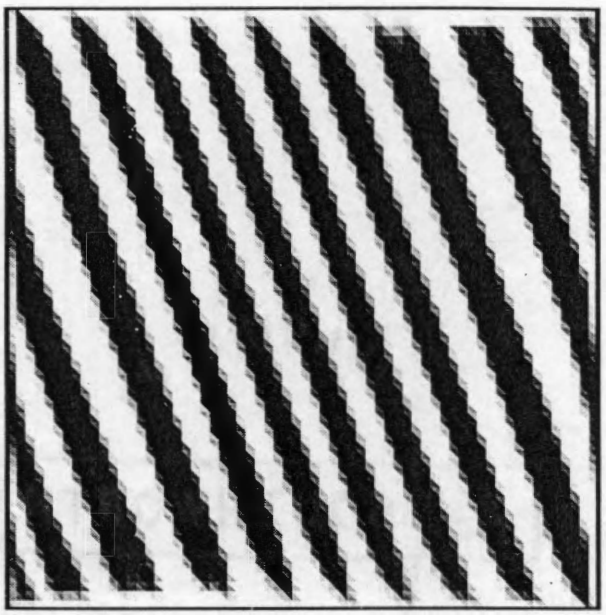

Figure 4.6.6 Twinning data with Q = 99.

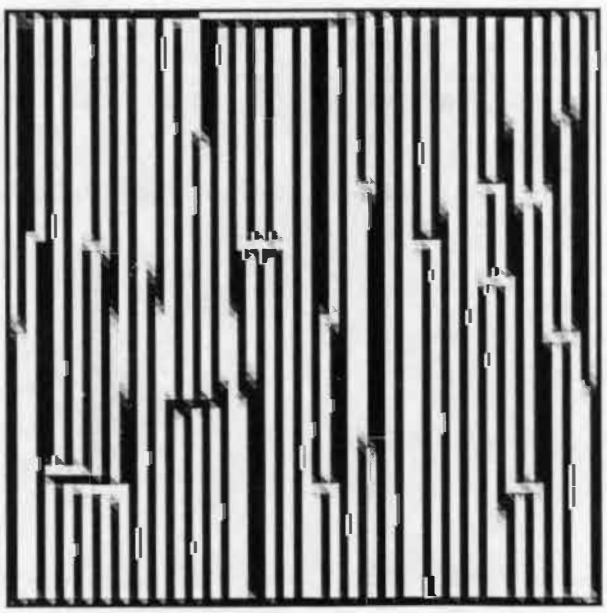

Figure 4.6.7 Twinning data with Q = 56.

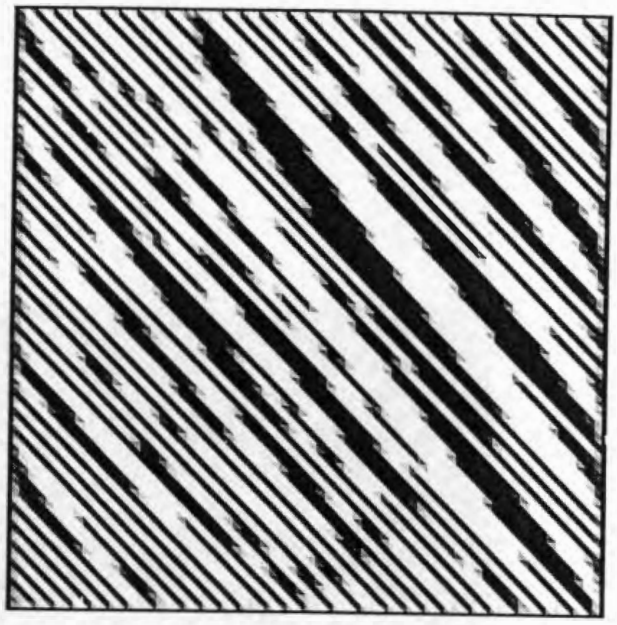

Figure 4.6.8 Twinning data with Q = 31.

Figure 4.6.9 Twinning data with Q = 52.

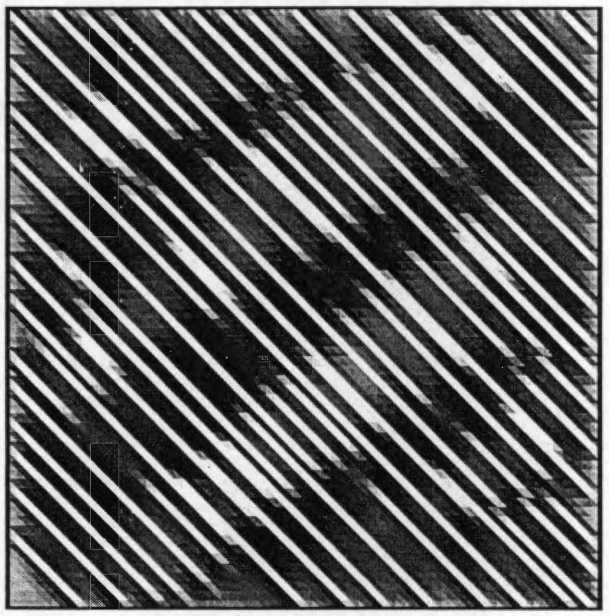

Figure 4.6.10 Twinning data with Q = 82.

Conclusion

The purpose of analyzing twinning data was to attempt to find a numeric measure of quality. We wrote the final formula for twinning quality with regard to uniformity in frequency and norm.

The DWT was an excellent tool for detecting phase changes in twinning data when $\lambda_m \geq 25$. However, for frequencies below 25 cycles we were unable to use the DWT coefficients $\{c_{1,n}^d\}$, $n = 1,...,33$; the DWT algorithm in Section 4.4 produced false results since the sign of $\{c_{1,n}^d\}$ changed with f, not with phase change. Other levels of analysis were difficult to interpret; no clear pattern between DWT coefficients $\{c_{m,n}^d\}$ for $m > 1$ and f exits that could be used in a numerical algorithm. Consequently, the quality number could be too low for data with frequency $\lambda_m \geq 25$ since phase change data lowers the Q-number.

The statistics we collect and analyze on λ_m are useful but flaws exist in frequency detection. For example, the DFT along with Algorithms 4.3 and 4.4 gives excellent frequency results unless data had a graph similar to the graph in Figure 4.6.5.

More research is needed to find what success the Q-number has in measuring twinning quality. Twinning data comes from the Finite Element Method where we fix the mesh size in the reference domain to be $h_{64} = \frac{1}{64}$. This limits the frequencies we detect in twinned material to $\lambda_m \leq 32$ with clusters of frequencies near $\lambda_m = 4$, $\lambda_m = 8$, $\lambda_m = 16$ and $\lambda_m = 32$. If the Q-number is adequate it should accurately measure uniformity and improve the methods used to find approximate twinned configurations.

Bibliography

Aboufadel, Edward and Schlicker, Steven. Discovering Wavelets. New York: John Wiley & Sons, Inc., 1999.

Cartwright, Mark. Fourier Methods for Mathematicians, Scientists and Engineers. Chichester, West Sussex, England: Ellis Horwood Limited, 1990.

Collins, Charles Robert, "Computation and Analysis of Twinning in Crystalline Solids" (Ph.D dissertation, University of Minnesota, 1990.)

Daubechie, Ingrid. Ten Lectures on Wavelets. Philadelphia: Society for Industrial and Applied Mathematics, 1992.

Davis, Harry F. Fourier Series and Orthogonal Functions. New York: Dover Publications, Inc., 1963.

Hanselman, Duane and Bruce Littlefield. The Student Edition of Matlab. Upper Saddle River: Prentice Hall, 1997.

Rudin, Walter. Principles of Mathematical Analysis. New York: McGraw-Hill Book Company, 1976.

Appendix

A.1 MODU

```
function [v1,v2,G] = modu(u1,u2)
    % converts (u1,u2) from deformations to displacements
    % using the boundary conditions (assumed linear)

    % Input:
       % matrices u1 and u2 from the program data.m
    % Output:
       % matrices v1 and v2 with linear trends removed

    % crc: 7/13/99

    [m,n] = size(u1);

    % assume the BC are u = Fx, so find 2x2 matrix F from the corners

    F(1,1) = u1(1,m);
    F(1,2) = u1(m,1);
    F(2,1) = u2(1,m);
    F(2,2) = u2(m,1);

    x = [0:m-1]/(m-1);
    [X,Y] = meshgrid(x);

    v1 = u1 - F(1,1)*X - F(1,2)*Y;
    v2 = u2 - F(2,1)*X - F(2,2)*Y;

    if (nargout>2)
       G = F;
    end
```

A.2 MIN2ND

```
function [w1,w2,th] = min2nd(v1,v2)
    % finds the angle th so that w2 = sin(th)*v1+cos(th)*v2
    % has minimum norm

    A = sum(sum(v1.^2));
    B = sum(sum(v1.*v2));
    C = sum(sum(v2.^2));

    th = atan2(2*B,(C-A))/2;
    w2 = sin(th)*v1 + cos(th)*v2;

    nm1 = sum(sum(w2.^2));

    th = th + pi/2;
    w2 = sin(th)*v1 + cos(th)*v2;

    nm2 = sum(sum(w2.^2));


    if (nm1<nm2)
       th = th - pi/2;
       w2 = sin(th)*v1 + cos(th)*v2;
    end

        w1 = cos(th)*v1 - sin(th)*v2;
```

A.3 F2M

```
function f2 = f2m(w)
% Filter for rows or columns of data matrix.
% Finds set of gamma_k's each row or column.
% Input:
    % Data array.
% Output:
    % Array of gamma_k's for each row or column.

f2=[];
[n,m] = size(w);
for i = 1:m
   y = w(:,i);
   z = fft(y,n-1); % n-point fft, padded if less, truncated
                   % if more.

   dy = diff(y(1:n-1)); % DIFF(X), for a vector X,
                        % is [X(2)-X(1)  X(3)-X(2) ...
                        % X(n)-X(n-1)].

   for k=2:(n-1)/2
       x = zeros(n-1,1);  % ZEROS(M,N) or ZEROS([M,N])
                          % is an M-by-N matrix of zeros.
       x(k) = z(k);          % This and next statement arrange
       x(n+1-k) = z(n+1-k);  % coefficients in correct order
        x = real(ifft(x));
        f2(i,k) = sum(diff(x).*dy);
   end
end
```

A.4 DFREQ

```
function [k,c] = dfreq(y)
% Takes data, finds dominant trigonometric frequency, gamma_j, for set of rows or
columns,
% WDT 11/17/00
% Input:
    % Data array
% Output:
    % Dominant trigonometric frequency.
fm=f2m(y);  %find dominant frequencies
[fmax,k]=max(fm);
fty=fft(y);  % find fourier coefficients
c=fty(k);
k;
c;
```

A.5 SWTHCOND

```
function [k,d,lamda]=swthcond(y)
% Fits sawtooth wave to data.
% Input:
    % One by N or N by one data array.
% Output:
    % Amplitude, frequency and phase angle of triangle wave fit to data.

[m,n]=size(y);
y=y(:)';
```

```
x=[0:n]/n;
x=x(1:n);

[k1,c1]=dfreq2(y');
theta=angle(c1);
lamda=4.03545869807927*k1-4.79639233108000;  % Equation found from
                                             % linear regression.


b=2+.5*mod(lamda-2,4);
a=-2/pi;
b=(1/lamda)*(a*theta+b); % Need scaling 1/lamda to adjust phase angle
                         % to find freq.

scl=(norm(y))/(norm(sawth(lamda*x-b)));
q=[lamda,b,scl];
[k,d]=fminsearch('funtest',q',[],x,y); %d is value of objective function

if abs((4.79639233108000+k(1))/4.03545869807927-k1)>0.5  % fminsearch
                  % finds triangle wave freq. represented by k(1)
   k(1)=lamda;   % if k(1) too far from k1, fminsearch did not work.
end
```

A.6 FUNTEST

```
function [d]=funtest(p,x,y)
% function to be minimized by fminsearch in swthcond.
lamda=p(1);a=p(2);scl=p(3);
e=scl*sawth(lamda*x-a);

d=norm(y-e);
```

A.7 WVLTANAL

```
function pshftnum = wvltanal(y)
% Input:
    % y is row or column of w1 or w2 data from snp.m
% Output:
    % pshftnum is number of phaseshifts for data with frequency, lamda,
    % greater than 25 cycles.

[c,l]=wavedec(y,4,'db2');  % Daubechie-4 wavelet, 4 levels of analysis.
pshftnum=find(abs(diff(sign(c(41:73))))>=1);  % Finds when wvlt coeff
                                              % switch sign.
                                              % Works as phaseshift indicater
                                              % when frequency is greater than
                                              % 25 cycles.
```

A.8 DANALYZE2

```
function [Q,af,ml,mi,m,fnd,A,bndrs]=danalyze2(data)

% Input:
    % Twinning data from Matlab file.
% Output:
    % Q is array of quality numbers from different formulas,
```

```
    % af is average freq. of all freq. within .5 cycles of the mode.
    % ml is mode length which gives the number of rows or columns of
    %    data with frequency within .5 cycles of mode frequency.
    % mi is an array   of mode indices.
    % m is the first two freq. modes in the array
    % fnd freq., norm data.
    % A is an array of optimal frequencies found for each row or column
    % of the data array.

[w1,w2,th]=cond2(data);
[k,l]=size(w1);

x=[0:64]/64;
x=x(1:64);


[anw1r,normw1r]=nrmd(w1,k);  % Average norm of w* rows, norm of w* rows
[anw1c,normw1c]=nrmd(w1',k); % Likewise for columns
[anw2r,normw2r]=nrmd(w2,k);
[anw2c,normw2c]=nrmd(w2',k);

if (anw1r+anw1c)>=(anw2r+anw2c)

   [normw1rs,spectrumw1r,phshftw1r,pshftnumw1r]=snp(w1,x,k); %spectrum,
   [normw1cs,spectrumw1c,phshftw1c,pshftnumw1c]=snp(w1',x,k);
   pshftnumw1ra=sum(pshftnumw1r)/k; %average of pshftnumw1r's
   pshftnumw1ca=sum(pshftnumw1c)/k;
   [ml1r,miw1r,afw1r,amfw1r,mfw1r,m1w1r,m2w1r,sfw1r,snw1r,snw1rs,aon1rs]=
                                              dstat(spectrumw1r,n    ormw1r,normw1rs);

[ml1c,miw1c,afw1c,amfw1c,mfw1c,m1w1c,m2w1c,sfw1c,snw1c,snw1cs,aon1cs]=dstat(spectrumw1c,n
                                                                    ormw1c,normw1cs);

   [mxf,d]=max([afw1r,afw1c]); % This and the next line find max and min of the average
   [mnf,e]=min([afw1r,afw1c]); % frequency of data rows or columns within .5 cycles of
                               % the mode.
   a=atan(mnf/mxf)*180/pi; % Gives approximate angle in degrees.

   if (a<=20)&(d==1)

[ml1c,miw1c,afw1c,amfw1c,mfw1c,m1w1c,m2w1c,sfw1c,snw1c,snw1cs,aon1cs]=dstat(spectrumw1c,n
                                                                ormw1c,normw1cs,1);
      % These output variables include mode 1 and mode 2 of frequencies.
      bw1r=ntcons(miw1r,k); % b = # of borders between data near mode frequency.
      bw1c=ntcons(miw1c,l);
      fndw1r=[sfw1r,snw1rs,aon1rs]; % Freq., norm data, w1r.
      fndw1c=[sfw1c,snw1cs,aon1cs];

      anw1=(anw1r+anw1c)/2; % average norm of w1 over rows and columns


Q=qnums(k,amfw1r,amfw1c,sfw1r,sfw1c,ml1r,ml1c,aon1rs,aon1cs,bw1r,bw1c,pshftnumw1ra,pshftn
umw1ca,2);

   elseif (a<20)&(d==2)

[ml1r,miw1r,afw1r,amfw1r,mfw1r,m1w1r,m2w1r,sfw1r,snw1r,snw1rs,aon1rs]=dstat(spectrumw1r,n
ormw1r,normw1rs,1);
      % Same comment as for w1c data above except for w1r data this time.
      bw1r=ntcons(miw1r,k); % b = # of borders between data near mode frequency.
      bw1c=ntcons(miw1c,l);
      fndw1r=[sfw1r,snw1rs,aon1rs]; % Freq., norm data, w1r.
      fndw1c=[sfw1c,snw1cs,aon1cs];

      anw1=(anw1r+anw1c)/2; % average norm of w1 over rows and columns


Q=qnums(k,amfw1r,amfw1c,sfw1r,sfw1c,ml1r,ml1c,aon1rs,aon1cs,bw1r,bw1c,pshftnumw1ra,pshftn
umw1ca,3);
```

```
   else
      bw1r=ntcons(miw1r,k) % b = # of borders between data near mode frequency.
      bw1c=ntcons(miw1c,1)
      fndw1r=[sfw1r,snw1rs,aon1rs]; % Freq., norm data, w1r.
      fndw1c=[sfw1c,snw1cs,aon1cs];

      anw1=(anw1r+anw1c)/2; % average norm of w1 over rows and columns


Q=qnums(k,amfw1r,amfw1c,sfw1r,sfw1c,ml1r,ml1c,aon1rs,aon1cs,bw1r,bw1c,pshftnumw1ra,pshftn
umw1ca,1);

   end


   bndrs=[bw1r bw1c];
   A(1,1:ml1r)=mfw1r;A(2,1:ml1c)=mfw1c; % Optimal freq.
   fnd=[fndw1r;fndw1c]; % Freq. norm data
   ml=[ml1r,ml1c]; % Number of entries near mode
   m=[m1w1r,m2w1r;m1w1c,m2w1c]; % rounded mode1 and mode2 data
   af=[afw1r,afw1c];
   mi=1;mi(2,1:ml1r)=miw1r;mi(3,1:ml1c)=miw1c;

elseif (anw1r+anw1c)<(anw2r+anw2c)

   [normw2rs,spectrumw2r,phshftw2r,pshftnumw2r]=snp(w2,x,k);
   [normw2cs,spectrumw2c,phshftw2c,pshftnumw2c]=snp(w2',x,k);
   pshftnumw2ra=sum(pshftnumw2r)/k;
   pshftnumw2ca=sum(pshftnumw2c)/k;

[ml2r,miw2r,afw2r,amfw2r,mfw2r,m1w2r,m2w2r,sfw2r,snw2r,snw2rs,aon2rs]=dstat(spectrumw2r,n
                                                                    ormw2r,normw2rs);

[ml2c,miw2c,afw2c,amfw2c,mfw2c,m1w2c,m2w2c,sfw2c,snw2c,snw2cs,aon2cs]=dstat(spectrumw2c,n
                                                                    ormw2c,normw2cs);

[mxf,d]=max([afw2r,afw2c]); % This and the next line find max and min of the average
[mnf,e]=min([afw2r,afw2c]); % Frequency of data rows or columns within .5 cycles of the
                           % mode.
   a=atan(mnf/mxf)*180/pi; % Gives approximate angle in degrees to adjust mode of
                           % analysis.  Small angles associated with high frequencies.

   if (a<=20)&(d==1)

[ml2c,miw2c,afw2c,amfw2c,mfw2c,m1w2c,m2w2c,sfw2c,snw2c,snw2cs,aon2cs]=dstat(spectrumw2c,n
ormw2c,normw2cs,1);
      % These output variables are updated to include mode 1 and mode 2 of frequencies.
      bw2r=ntcons(miw2r,k)
      bw2c=ntcons(miw2c,1)

      fndw2r=[sfw2r,snw2rs,aon2rs];% Freq., norm data of w2 rows
      fndw2c=[sfw2c,snw2cs,aon2cs];

      anw2=(anw2r+anw2c)/2; % average norm of w2 over rows and columns


Q=qnums(k,amfw2r,amfw2c,sfw2r,sfw2c,ml2r,ml2c,aon2rs,aon2cs,bw2r,bw2c,pshftnumw2ra,pshftn
umw2ca,2);

   elseif (a<20)&(d==2)


[ml2r,miw2r,afw2r,amfw2r,mfw2r,m1w2r,m2w2r,sfw2r,snw2r,snw2rs,aon2rs]=dstat(spectrumw2r,n
ormw2r,normw2rs,1);
      % Same comment as for w2c data above except for w2r data this time.
      bw2r=ntcons(miw2r,k)
      bw2c=ntcons(miw2c,1)

      fndw2r=[sfw2r,snw2rs,aon2rs];% Freq., norm data of w2 rows
      fndw2c=[sfw2c,snw2cs,aon2cs];
```

```
      anw2=(anw2r+anw2c)/2; % average norm of w2 over rows and columns


Q=qnums(k,amfw2r,amfw2c,sfw2r,sfw2c,ml2r,ml2c,aon2rs,aon2cs,bw2r,bw2c,pshftnumw2ra,pshftn
ums2ca,3);

   else
      bw2r=ntcons(miw2r,k) % b = # of borders between data near mode frequency.
      bw2c=ntcons(miw2c,1)
      fndw2r=[sfw2r,snw2rs,aon2rs]; % Freq., norm data, w2r.
      fndw2c=[sfw2c,snw2cs,aon2cs];

      anw2=(anw2r+anw2c)/2; % average norm of w2 over rows and columns


Q=qnums(k,amfw2r,amfw2c,sfw2r,sfw2c,ml2r,ml2c,aon2rs,aon2cs,bw2r,bw2c,pshftnumw2ra,pshftn
umw2ca,1);

   end


   bndrs=[bw2r bw2c];
   A(1,1:ml2r)=mfw2r;A(2,1:ml2c)=mfw2c;
   fnd=[fndw2r;fndw2c];
   ml=[ml2r,ml2c];
   m=[m1w2r,m2w2r;m1w2c,m2w2c]; % rounded mode1 and mode2 data
   af=[afw2r,afw2c];
   mi=2;mi(2,1:ml2r)=miw2r;mi(3,1:ml2c)=miw2c;
end
```

A.9 DSTAT

```
function [ml,mi,af,amf,mf,m1,m2,sf,sn,sns,aon]=dstat(s,n,ns,op)

% Provide statistics for frequency, mode and norm data.

% Use op=1 to evaluate high frequency data.

% Arguments: s = spectrum or the frequency band of the rows and columns in the
% data, n = norm of data rows or columns, and ns = norm of sawtooth solution.

% Output:
    % ml = mode length.
    % mi = mode indices.
    % af = average frequency over all rows (or columns).
    % amf = average of frequencies within 0.5 cycles of modal freq.,
    % m1 = mode 1 or frequency occuring most often, m2 = mode 2.
    % sf = standard deviation of frequency.
    % sn = standard deviation of norm of data.
    % sns = standard deviation of norms of sawtooth solutions.
    % aon = average norm of data rows (or columns) with frequency nearest to that of modal
    % frequency.


% Find frequency modes and build arrays with indices of rows (or columns)
% within 0.5 cycles of modal frequencies.

rf=round(s/4); % Rounded towards nearest integer.  s/4 since freq of triangle waves needs
               % scaling by four for appropriate frequency.
[m1,m2]=mode2(rf); % m1 and m2 are 1st and 2nd frequency modes

if nargin==3
   if (m1-2^(log2(m1)-2)<m2) & (m2<m1+2^(log2(m1)-2)) % The appropriate frequency
                                                      % neighborhood is set around
```

```
                                                    % modal frequency.
      mi=find((rf==m1)|(rf==m2)); % Modal index, mi, is the index of rounded
                                  % frequencies, rf, equal to the modal frequency.
   else
      mi=find(rf==m1);
   end
elseif op==1                    % The result of using op==1 is that the mode
   mi=find((rf==m1)|(rf==m2)); % length, ml, includes two modes and has more
end                            % entries.  This adjusts for inaccuracies in
                               % detecting frequencies in data with small
                               % angles with either rows or columns having a
                               % high frequency and improves the accuracy of
                               % the final quality numbers by eliminating
                               % noise from frequency measurements.


% Index arrays are used to find statistical data below.

f=s/4; % Frequency of optimized sawtooth waves.  Divided by four since sawtooth
       % waves have a natural frequency of four cycles for each cycle.
sf=std(f); % Standard deviation of sawtooth wave frequencies optimized to fit
           % data.
mf=f(mi); % Optimal frequencies that were rounded toward the mode.
amf=sum(mf,2)/length(mi); % Average frequency of optimized solutions within .5
                          % cycles mf the mode.
af=sum(f)/length(f); % Average frequency of optimized solutions.

ons=ns(mi); %Norm (optimal) of sawtooth solutions within 0.5 of frequency mode.
on=n(mi); %Norm (optimal) of data matching index of data with frequency mode.

aon=sum(on)/length(mi); % Average of norms of optimal sawtooth solutions.
sns=std(ons); % Standard deviation of norm of sawtooth solutions.
sn=std(n); % Standard deviation of norms from rows or columns in data.
ml=length(mi);
```

A.10 NTCONS

```
function n=ntcons(v,k)

% Input:
    % v = string of integers, in this case index numbers for data
    % closest to optimal solution.
    % k = longest possible string.
% Output:
    % n = number of breaks in a string of integers implicitly starting
    % with n=1 and ending with k.
w=1:k;
x=setdiff(w,v);
ind=find(diff(x)>1);
if isempty(x)==1
   n=length(ind);
else
   n=length(ind)+1;
end
```

A.11 QNUMS

```
function Q=qnums(k,amfr,amfc,sfr,sfc,mlr,mlc,aonr,aonc,br,bc,psnra,psnca,op)
% r=row data, c=column data, k=length of data row or column.
% op=1 then full formula, op=2 then sfc and bc =0, op=3 then sfr and br = 0.
```

```
% psnra = pshftnumrow average, psnca = pshftnumcolumn average
if op==1

qa=((mlr/(k+(sfr+br+psnra)/amfr))*aonr+(mlc/(k+(sfc+bc+psnca)/amfc))*aonc)/(aonr+aonc);

elseif op==2
   qa=((mlr/(k+(sfr+br+psnra)/amfr))*aonr+(mlc/(k))*aonc)/(aonr+aonc);

elseif op==3
   qa=((mlr/(k))*aonr+(mlc/(k+(sfc+bc+psnca)/amfc))*aonc)/(aonr+aonc);
end

Q=[qa];
```

A.12 SNP

```
function [ns,s,p,pshftnum]=snp(D,x,m)

% Spectrum, norm, phaseshift analysis of data, D. Size of data is in square arrays of
size
% mxm.  Output variables are ns = norm of optimized sawth solution,
% s = spectrum, p = phaseshift.
ns=[];s=[];p=[];pshftnum=[];

for i=1:m
   y=D(i,:); % D = data
   [k,d,lamda]=swthcond(y);
   s=[s k(1)];  %k(1)is the frequency of optimal sawtooth function
   p=[p k(2)];  %k(2) is the phaseshift of optimal sawtooth function
   %ampw1r=[ampw1r k(3)];  %k(3) is the amplitude of optimal sawtooth function
   ns=[ns norm(k(3)*sawth(k(1)*x-k(2)))];
   if k(1)>=100
      psnum=wvltanal(y);
         else
      psnum=0;
   end
   pshftnum=[pshftnum psnum];
end
```

Vita

Dean Turley graduated from Berea College with a Bachelor of Arts degree in Mathematics and Philosophy in 1984. He worked as Adjunct Instructor for Roane State Community College and Pellissippi State Technical Community College until the fall semester of 2000. In 2002 he finished his MS in Mathematics at the University of Tennessee, Knoxville.