# YOUR TITLE

BY

YOUR NAME HERE

A Thesis Submitted to the Graduate Faculty of

WAKE FOREST UNIVERSITY GRADUATE SCHOOL OF ARTS AND SCIENCES

in Partial Fulfillment of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Statistical Sciences

Date goes here

Winston-Salem, North Carolina

Approved By:

Your Advisor, Ph.D., Advisor

Committee Chair, Ph.D., Chair

Committee member, Ph.D.

Committee member, Ph.D.

# Acknowledgments

There are many people who helped to make this thesis possible. ......

# Table of Contents

# Abstract

Your Name

Your abstract text goes here

2nd Paragraph can go here if needed

and a 3rd

# Chapter 1: Introduction

This is sample text from an actual Thesis:

As images pass through the earth's atmosphere they become distorted. In some cases so much so that the image has no similarities to it's "actual" form. The wave front relinquishes it's form for a variety of reasons. Some of which include humidity, air temperature and general atmospheric turbulence [7]. By definition adaptive optics is the scientific and engineering discipline which deals with the control of light in some form of an adaptive looping process [7]. Even though no single person may be accredited as the founder of adaptive optics there have been vast advancements made in the past 50 years. The basic process behind adaptive optics is to obtain a sample of the gradient of the incoming wavefront which may be accomplished through the use of a Shack-Hartmann wave front sensor when a point source is available. Perhaps the most common method for correcting atmospheric aberrations is by phase conjugation. This method is the focal point in adaptive optics. Next, the obtained data is used to correct the wave front.

When any wave front passes through some aberrator it is modified. Once the distorted wave front bounces off of a conventional uncorrected mirror it again passes through the aberrator and the distortion in the wave front is now doubled as seen in Figure 1 located in the Appendix. The magnititudes of the bumps in the final distorted wave front directly determine the mirror correction necessary in order to retrieve the original wavefront. Multiplication of every distortion by - 1/2 determines the amount of correction needed over each particular mirror segment at the position in which the wavefront reflects off of the mirror segment. This is a very tricky process since most aberrations are constantly changing and therefore the correction for them must be a continuous process and must repeatedly occur in order to achieve accurate results. Correction is accomplished by quickly positioning the deformable mirror to correct the wavefront. The detection, estimation and correction process should not take longer than one one-hunderdth of a second [7]. Since the goal is to obtain real time data. The final images may be further enhanced by post-processing methods, but that topic will not be discussed in this paper.

[remainder of chapter deleted]

# Chapter 2: Name of chapter 2

In the mid 1960's, the Fast Fourier Transform, or FFT, was born. By 1977, various methods of FFT's were implemented in a wide variety of applications. It is still considered to be one of greatest advancements in Numerical Analysis. FFT algorithms were developed as a "quick" method for computing the Discrete Fourier Transform. The Discrete Fourier Transform, or DFT, is the result obtained by discretizing the Fourier Series. It is important to remember that the function in consideration must have periodicity, a finite number of maxima and minima during any period, a finite number of discontinuities if there are any at all, and it must be absolutely integrable in any period in order to have a fourier series representation [5]. Any algorithm whose computational efficiency is on the order of N $log_2$N is generally considered an FFT method. This implies that solutions to the inverse Discrete Fourier Transforms are also considered FFTs.

There are many types of FFT algorithms, but they fall into one of two general categories. The decimation in time approach and the decimation in frequency approach. The decimation in time approach breaks apart the time dependent input vector x in order to approximate the solution. The decimation in frequency approach breaks apart the frequency dependent vector y in its computation.

In order to use the so-called Radix-2 FFT, the matrix in question must be n × n where $n = 2^i$ ∃ i ≥ 1. The general Radix algorithm requires the n × n matrix to be highly composite (i.e. having many small factors). There are also algorithms for computing FFTs of non-square matrices. However, the Radix-2 algorithms allow for faster execution than the more general ones. It is possible to modify a matrix whose dimension is not a power of two by padding it with zeros to obtain a square n × n matrix where n is some power of two.

Let the vector $y = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix}$ be the Discrete Fourier Transform of $x = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix}$.

Each $y_k = \sum_{j=0}^{n-1}[cos(2\pi kj/n) - isin(2\pi kj/n)]x_j = \sum_{j=0}^{n-1} e^{-2\pi kji/n}x_j$ for k = 0, 1, ...,n-1. There are several variations on computing the Radix-2 FFT. One is the omition of the $\frac{1}{\text{n}}$ term for each $y_k$. Another is to omit the negative sign from the power of the exponential function. The DFT of y may also be seen as the product of a matrix and a vector where y = $F_n$x and $F_n$ is an n × n matrix with rows indexed by k and columns indexed by j. Each entry in $F_n$ is denoted by f(k,j) and is equal to $e^{-2\pi kji/n}$. In order to reduce the number of flops the convention of dropping the 1/n term will be applied in the formulation of the DFT.

Sample Matlab code to set up the Discrete Fourier Transform matrix given by [8]:

```
for k = 0:n-1
    m = $e^{-2\pi/n}$
    F(k,0) = 1
    for j = 1:n-1
    F(k,j) = mF(k,j-1)
    end
end
```

The efficiency of an algorithms is commonly measured in flops which are floating point operations. There are two flops for complex addition and six flops for complex multiplication. The total number of flops for the above code is given by O($6n^2$) flops.

Assuming the vector x still has length n. Define

$$\omega_m = diag(1, e^{-2\pi/n}, (e^{-2\pi/n})^2, ..., (e^{-2\pi/n})^(m-1)) \tag{2.1}$$

where m is initially $\frac{\mathrm{n}}{2}$ and diag refers to an m × m matrix with the entries listed going down the diagonal. Define $EOPerm_n$ to be an n × n matrix using sequential even and odd format respectively and beginning with 0 as the first element in the even list. One of the interesting properties of $EOPerm_n$ is that the transpose of this matrix is its inverse such that $EOPerm_n EOPerm_n^T = I_n$.

Applying $EOPerm_n$ to $F_n$ on the right will yield a matrix of the form

$$F_n EOPerm_n = \begin{pmatrix} F_{\frac{\mathrm{n}}{2}} & \omega_{\frac{\mathrm{n}}{2}} F_{\frac{\mathrm{n}}{2}} \\ F_{\frac{\mathrm{n}}{2}} & -\omega_{\frac{\mathrm{n}}{2}} F_{\frac{\mathrm{n}}{2}} \end{pmatrix} = \begin{pmatrix} I_{\frac{\mathrm{n}}{2}} & \omega_{\frac{\mathrm{n}}{2}} \\ I_{\frac{\mathrm{n}}{2}} & -\omega_{\frac{\mathrm{n}}{2}} \end{pmatrix} \begin{pmatrix} F_{\frac{\mathrm{n}}{2}} & 0 \\ F_{\frac{\mathrm{n}}{2}} & 0 \end{pmatrix}$$

Since $EOPerm_n^{-1} = EOPerm_n^T$ then multiplying both sides by $EOPerm_n)^T$ on the right yields:

$$F_n EOPerm_n EOPerm_n^T = \begin{pmatrix} I_{\frac{\mathrm{n}}{2}} & \omega_{\frac{\mathrm{n}}{2}} \\ I_{\frac{\mathrm{n}}{2}} & -\omega_{\frac{\mathrm{n}}{2}} \end{pmatrix} \begin{pmatrix} F_{\frac{\mathrm{n}}{2}} & 0 \\ F_{\frac{\mathrm{n}}{2}} & 0 \end{pmatrix} EOPerm_n^T$$

$$F_n I_n = F_n = \begin{pmatrix} I_{\frac{\mathrm{n}}{2}} & \omega_{\frac{\mathrm{n}}{2}} \\ I_{\frac{\mathrm{n}}{2}} & -\omega_{\frac{\mathrm{n}}{2}} \end{pmatrix} \begin{pmatrix} F_{\frac{\mathrm{n}}{2}} & 0 \\ F_{\frac{\mathrm{n}}{2}} & 0 \end{pmatrix} (EOPerm_n)^T$$

Multiplying both sides by the vector x on the right gives:

$$F_n x = \begin{pmatrix} I_{\frac{\mathrm{n}}{2}} & \omega_{\frac{\mathrm{n}}{2}} \\ I_{\frac{\mathrm{n}}{2}} & -\omega_{\frac{\mathrm{n}}{2}} \end{pmatrix} \begin{pmatrix} F_{\frac{\mathrm{n}}{2}} & 0 \\ F_{\frac{\mathrm{n}}{2}} & 0 \end{pmatrix} x(EOPerm_n)^T$$

$$F_n x = \begin{pmatrix} I_{\frac{\mathrm{n}}{2}} & \omega_{\frac{\mathrm{n}}{2}} \\ I_{\frac{\mathrm{n}}{2}} & -\omega_{\frac{\mathrm{n}}{2}} \end{pmatrix} \begin{pmatrix} F_{\frac{\mathrm{n}}{2}} & 0 \\ F_{\frac{\mathrm{n}}{2}} & 0 \end{pmatrix} \begin{pmatrix} x(0:2:n-2) \\ x(1:2:n-1) \end{pmatrix}$$

$$y = F_n x = \begin{pmatrix} I_{\frac{\mathrm{n}}{2}} & \omega_{\frac{\mathrm{n}}{2}} \\ I_{\frac{\mathrm{n}}{2}} & -\omega_{\frac{\mathrm{n}}{2}} \end{pmatrix} \begin{pmatrix} F_{\frac{\mathrm{n}}{2}} x(0:2:n-2) \\ F_{\frac{\mathrm{n}}{2}} x(1:2:n-1) \end{pmatrix}$$

Repeating these steps for C = $\frac{\mathrm{n}}{2} F_C x(0:2:n-2)$ and $F_C x(1:2:n-1)$ follows similarly. This process can be repeated until c = 1 when n is some power of two.

Since the Fast Fourier Transform is a "quick" solution to the DFT, then the number of flops used to compute y should be much less than an n × n matrix * n vector, which would be $8n^2$ flops for complex numbers.

The FFT method utilizes the special structure of the DFT matrix. The basic concept of an FFT involves systematically working backwards. Assuming our original modified matrix is n = $2^t$ there are t + 1 computation levels and our current computational level is s which is initially s = 0 working backwards. The initial step realizes that $F_1 x_j = x_j$ and $F_1 x_k = x_k$ since $F_1 = 1$. Given two DFT matrices that are to be combined are a and b each with length $\frac{\mathrm{n}}{2}$ then

$$\begin{aligned} a &= F_{\frac{\mathrm{c}}{2}} x(k : \frac{\mathrm{n}}{2^{\mathrm{s}-1}} : n-1) && (2.2) \\ b &= F_{\frac{\mathrm{c}}{2}} x(k + \frac{\mathrm{n}}{2^{\mathrm{s}-1}} : \frac{\mathrm{n}}{2^{\mathrm{s}-1}} : n-1) && (2.3) \end{aligned}$$

Using the original vector x, start at the $k^{th}$ position and include every $\frac{\mathrm{n}}{2^{\mathrm{s}-1}}$ entry until reaching the n-1 entry when forming the vector a and likewise for b. Also the $\omega_{\frac{\mathrm{c}}{2}}$ matrix defined earlier is needed, where n = $2^s$. Define $\zeta = \omega_{\frac{\mathrm{c}}{2}}$ b then we may obtain

$$F_c x(k : \frac{\mathrm{n}}{2^{\mathrm{s}}} : n-1) = \begin{pmatrix} I_{\frac{\mathrm{n}}{2}} & \omega_{\frac{\mathrm{n}}{2}} \\ I_{\frac{\mathrm{n}}{2}} & -\omega_{\frac{\mathrm{n}}{2}} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a+\zeta \\ a-\zeta \end{pmatrix} \tag{2.4}$$

[remainder of chapter deleted]

# Chapter 3: Chapter 3

Chapter text goes here

**Theorem 3.1.** *Matrix $B$ has a decomposition given by $B = \Theta E \lambda_B E^{-1} \Theta$ where $E$, $E^{-1}$, $\Theta$ and $\lambda_B$ are $N$ $X$ $N$ matrices containing real values.*

*Proof.* text of your proof....................text of your proof.................... text of your proof.................... text of your proof.................... text of your proof.................... text of your proof.................... text of your proof.................... text of your proof.................... text of your proof.................... □

**Corollary 1.** *If there is a Corollary to the Theorem*

**Definition 1.** *text for definition goes here*

# Chapter 4: Chapter 4

Text goes here

# Bibliography

[1] Angel, E. and R. Bellman Dynamic Programming and Partial Differential Equations. Academic Press, New York, 1972.

[2] Demmel, James W. Applied Numerical Linear Algebra. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1997.

[3] Golub, Gene H. and James M. Ortega Scientific Computing and Differential Equations. Academic Press, San Diego, California, 1992.

[4] Milman, Mark and Amir Fijany

[5] Ramirez, Robert W. The FFT, Fundamentals and Concepts. Prentice Hall Inc. , Englewood Cliffs, New Jersey, 1985.

[6] Walter A. Partial Differential Equations, An Introduction. John Wiley and Sons Inc. , New York. 1992.

[7] Tyson, Robert K. Principles of Adaptive Optics, Second Edition. Academic Press, San Diego, California, 1991.

[8] Van Loan, Charles Computational Frameworks for the Fast Fourier Transform. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1992.

[9] Walker, James S. Fast Fourier Transform, Second Edition. CRC Press Inc. , Boca Raton, Florida, 1996.

# Appendix A: A Title Goes Here

Text for the 1st Appendix

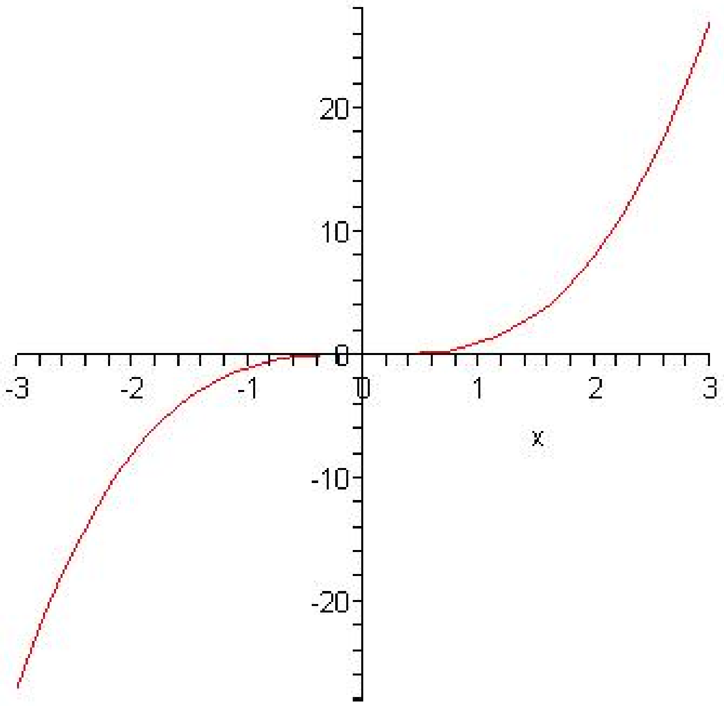
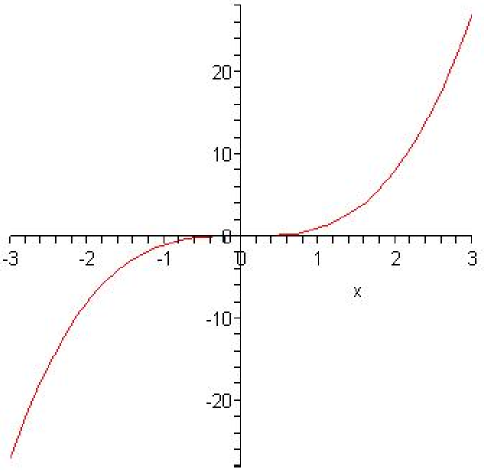


Figure A.1: Awesome Image


20
10
0
-10
-20
-3
-2
-1
0
1
2
3
x

# Appendix B: Title for 2nd Appendix

info goes here if needed