COMPUTATIONAL MATHEMATICS SERIES

INSIDE the FFT BLACK BOX

Serial and Parallel Fast Fourier Transform Algorithms

COMPUTATIONAL MATHEMATICS SERIES

INSIDE the FFT BLACK BOX

Serial and Parallel Fast Fourier Transform Algorithms

Eleanor Chu
University of Guelph
Ontario, Canada

Alan George
University of Waterloo
Ontario, Canada

CRC Press
Boca Raton London New York Washington, D.C.

Library of Congress Cataloging-in-Publication Data

Catalog record is available from the Library of Congress

International Standard Book Number 0-8493-0270-6
Library of Congress Card Number 99-048017
Printed in the United States of America 2 3 4 5 6 7 8 9 0
Printed on acid-free

Contents

Preface

The fast Fourier transform (FFT) algorithm, together with its many successful applications, represents one of the most important advancements in scientific and engineering computing in this century. The wide usage of computers has been instrumental in driving the study of the FFT, and a very large number of articles have been written about the algorithm over the past thirty years. Some of these articles describe modifications of the basic algorithm to make it more efficient or more applicable in various circumstances. Other work has focused on implementation issues, in particular, the development of parallel computers has spawned numerous articles about implementation of the FFT on multiprocessors. However, to many computing and engineering professionals, the large collection of serial and parallel algorithms remain hidden inside the FFT black box because: (1) coverage of the FFT in computing and engineering textbooks is usually brief, typically only a few pages are spent on the algorithmic aspects of the FFT; (2) cryptic and highly variable mathematical and algorithmic notation; (3) limited length of journal articles; and (4) important ideas and techniques in designing efficient algorithms are sometimes buried in software or hardware-implemented FFT programs, and not published in the open literature.

This book is intended to help rectify this situation. Our objective is to bring these numerous and varied ideas together in a common notational framework, and make the study of FFT an inviting and relatively painless task. In particular, the book employs a unified and systematic approach in developing the multitude of ideas and computing techniques employed by the FFT, and in so doing, it closes the gap between the often brief introduction in textbooks and the equally often intimidating treatments in the FFT literature. The unified notation and approach also facilitates the development of new parallel FFT algorithms in the book.

This book is self-contained at several levels. First, because the fast Fourier transform (FFT) is a fast "algorithm" for computing the discrete Fourier transform (DFT), an "algorithmic approach" is adopted throughout the book. To make the material fully accessible to readers who are not familiar with the design and analysis of computer algorithms, two appendices are given to provide necessary background. Second, with the help of examples and diagrams, the algorithms are explained in full. By exercising the appropriate notation in a consistent manner, the algorithms are explicitly connected to the mathematics underlying the FFT—this is often the "missing link" in the literature. The algorithms are presented in pseudo-code and a complexity analysis of each is provided.

Features of the book

• *The book is written to bridge the gap between textbooks and literature.* We believe this book is unique in this respect. The majority of textbooks largely focus on the underlying mathematical transform (DFT) and its applications, and only a small part is devoted to the FFT, which is a fast algorithm for computing the DFT.

• *The book teaches up-to-date computational techniques relevant to the FFT.* The book systematically and thoroughly reviews, explains, and unifies FFT ideas from journals across the disciplines of engineering, mathematics, and computer science from 1960 to 1999. In addition, the book contains several parallel FFT algorithms that are believed to be new.

• *Only background found in standard undergraduate mathematical science, computer science, or engineering curricula is required.* The notations used in the book are fully explained and demonstrated by examples. As a consequence, this book should make FFT literature accessible to senior undergraduates, graduate students, and computing professionals. The book should serve as a self-teaching guide for learning about the FFT. Also, many of the ideas discussed are of general importance in algorithm design and analysis, efficient numerical computation, and scientific programming for both serial or parallel computers.

Use of the book

It is expected that this book will be of interest and of use to senior undergraduate students, graduate students, computer scientists, numerical analysts, engineering professionals, specialists in parallel and distributed computing, and researchers working in computational mathematics in general.

The book also has potential as a supplementary text for undergraduate and graduate courses offered in mathematical science, computer science, and engineering programs. Specifically, it could be used for courses in scientific computation, numerical analysis, digital signal processing, the design and analysis of computer algorithms, parallel algorithms and architectures, parallel and distributed computing, and engineering courses treating the discrete Fourier transform and its applications.

Scope of the book

The book is organized into 24 chapters and 2 appendices. It contains 97 figures and 38 tables, as well as 25 algorithms presented in pseudo-code, along with numerous code segments. The bibliography contains more than 100 references dated from 1960 to 1999. The chapters are organized into three parts.

I. Preliminaries Part I presents a brief introduction to the discrete Fourier transform through a simple example involving trigonometric interpolation. This part is included to make the book self-contained. Some details about floating point arithmetic as it relates to FFT computation is also included in Part I.

II. Sequential FFT Algorithms This part contains fourteen relatively short chapters (3 through 16). Although the FFT, like binary search and quicksort, is commonly used in textbooks to illustrate the divide and conquer paradigm and recursive algorithms, the FFT has a unique feature: the application of the basic FFT algorithm

to "naturally ordered" input, if performed "in place," yields output in "bit-reversed" order. While this feature may be taken for granted by FFT insiders, it is often not addressed in detail in textbooks. Again, partly because of the lack of notation linking the underlying mathematics to the algorithm, and because it is understood by FFT professionals, this aspect of the FFT is either left unexplained or explained very briefly in the literature. This phenomenon, its consequences, and how to deal with it, is one of the topics of Part II.

Similarly, the basic FFT algorithm is generally introduced as most efficient when applied to vectors whose length N is a power of two, although it can be made even more efficient if N is a power of four, and even more so if it is a power of eight, and so on. These situations, as well as the case when N is arbitrary, are considered in Part II. Other special situations, such as when the input is real rather than complex, and various programming "tricks," are also considered in Part II, which concludes with a chapter on selected applications of FFT algorithms.

III. Parallel FFT Algorithms The last part deals with the many and varied issues that arise in implementing FFT algorithms on multiprocessor computers. Part III begins with a chapter that discusses the mapping of data to processors, because the designs of the parallel FFTs are mainly driven by data distribution, rather than by the way the processors are physically connected (through shared memory or by way of a communication network.) This is a feature not shared by parallel numerical algorithms in general.

Distributed-memory multiprocessors are discussed next, because implementing the algorithms on shared-memory architecture is straightforward. The hypercube multiprocessor architecture is particularly considered because it is so naturally compatible with the FFT algorithm. However, the material discussed later does not specifically depend on the hypercube architecture.

Following that, a series of chapters contains a large collection of parallel algorithms, including some that are believed to be new. All of the algorithms are described using a common notation that has been derived from one introduced in the literature. As in part II, dealing with the bit-reversal phenomenon is considered, along with balancing the computational load and avoiding communication congestion. The last two chapters deal with two-dimensional FFTs and the task of distributing the "twiddle factors" among the individual processors.

Appendix A contains basic information about efficient computation, together with some fundamentals on complexity notions and notation. Appendix B contains techniques that are helpful in solving recurrence equations. Since FFT algorithms are recursive, analysis of their complexity leads naturally to such equations.

Acknowledgments

This book resulted from our teaching and research activities at the University of Guelph and the University of Waterloo. We are grateful to both Universities for providing the environment in which to pursue these activities, and to the Natural Sciences and Engineering Research Council of Canada for our research support. At a personal level, Eleanor Chu owes a special debt of gratitude to her husband, Robert Hiscott, for his understanding, encouragement, and unwavering support.

We thank the reviewers of our book proposal and draft manuscript for their helpful suggestions and insightful comments which led to many improvements.

Our sincere thanks also go to Robert Stern (Publisher) and his staff at CRC Press for their enthusiastic support of this project.

Eleanor Chu
Guelph, Ontario

Alan George
Waterloo, Ontario

Part I

Preliminaries

Chapter 1

An Elementary Introduction to the Discrete Fourier Transform

This chapter is intended to provide a brief introduction to the discrete Fourier transform (DFT). It is not intended to be comprehensive; instead, through a simple example, it provides an illustration of how the computation that is the subject of this book arises, and how its results can be used. The DFT arises in a multitude of other contexts as well, and a dozen more DFT-related applications, together with information on a number of excellent references, are presented in Chapter 16 in Part II of this book. Readers familiar with the DFT may safely skip this chapter.

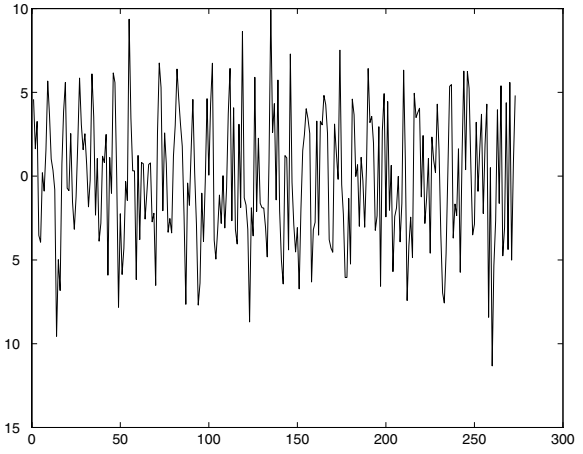
A major application of Fourier transforms is the analysis of a series of observations: $x_\ell, \ell = 0, \ldots, N-1$. Typically, N will be quite large: 10000 would not be unusual. The sources of such observations are many: ocean tidal records over many years, communication signals over many microseconds, stock prices over a few months, sonar signals over a few minutes, and so on. The assumption is that there are repeating patterns in the data that form *part* of the x_ℓ. However, usually there will be other phenomena which may not repeat, or repeat in a way that is not discernably cyclic. This is called "noise." The DFT helps to identify and quantify the cyclic phenomena. If a pattern repeats itself m times in the N observations, it is said to have *Fourier frequency* m.

To make this more specific, suppose one measures a signal from time $t = 0$ to $t = 680$ in steps of 2.5 seconds, giving 273 observations. The measurements might appear as shown in Figure 1.1. How does one make any sense out of it? As shown later, the DFT can help.

1.1 Complex Numbers

Effective computation of the DFT relies heavily on the use of complex numbers, so it is useful to review their basic properties. This material is elementary and probably well-known to most readers of this book, but it is included for completeness. Historically, complex numbers were introduced to deal with polynomial equations, such as $x^2 + 1 =$

Figure 1.1 Example of a noisy signal.



0, which have no real solutions. Informally, they can be defined as the set $\mathcal{C}$ of all "numbers" of the form $a + jb$ where a and b are real numbers and $j^2 = -1$.

Addition, subtraction, and multiplication are performed among complex numbers by treating them as binomials in the unknown j and using $j^2 = -1$ to simplify the result. Thus

$$(a + jb) + (c + jd) = (a + c) + j(b + d)$$

and

$$(a + jb) \times (c + jd) = (ac - bd) + j(ad + bc).$$

For the complex number $z = a + jb$, a is the *real part* of z and b is the *imaginary part* of z. The zero element of $\mathcal{C}$ is $0 + 0i$, and the additive inverse of $z = a + jb$ is $-a + i(-b)$. The multiplicative inverse z^{-1} is

$$z^{-1} = \frac{a - jb}{a^2 + b^2}.$$

The complex conjugate of $z = a + jb$ is denoted by $\bar{z}$ and is equal to $a - jb$. The *modulus* of z, denoted by $|z|$, is

$$\sqrt{z\bar{z}} = \sqrt{a^2 + b^2}.$$

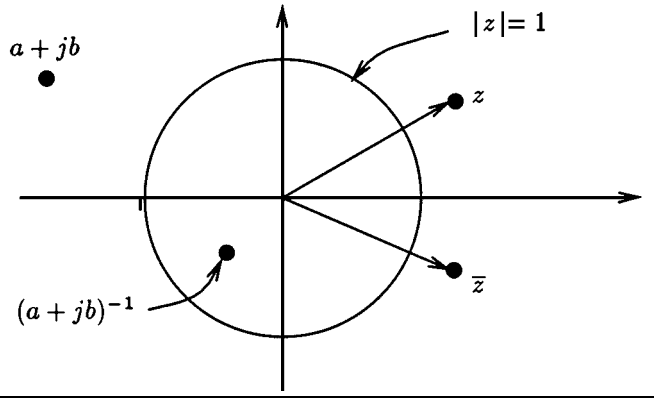
Some additional facts that will be used later are

$$e^z = e^{(a+jb)} = e^a e^{jb} \quad \text{and} \quad e^{jb} = \cos b + j \sin b.$$

Thus, $Re(e^z) = e^a \cos b$ and $Im(z) = e^a \sin b$.

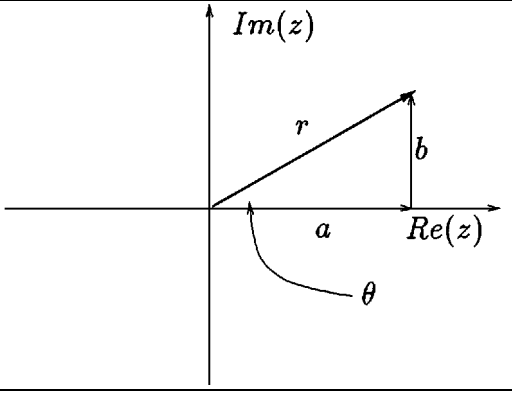
Just as a real number can be pictured as a point lying on a line, a complex number can be pictured as a point lying in a plane. With each complex number $a + jb$ one can associate a vector beginning at the origin and terminating at the point (a, b). These notions are depicted in Figure 1.2.

Figure 1.2 Visualizing complex numbers.



Instead of the pair (a, b), one can use the "length" (modulus) together with the angle the number makes with the real axis. Thus, $a + jb$ can be represented as $r\cos\theta + jr\sin\theta = re^{j\theta}$, where $r = |z| = \sqrt{a^2 + b^2}$ and $\theta = \arctan(b/a)$. This representation of a complex number is depicted in Figure 1.3.

Figure 1.3 Polar representation of a complex number.



Multiplication of complex numbers in polar form is straightforward: if $z_1 = a + jb = r_1 e^{j\theta_1}$ and $z_2 = c + jd = r_2 e^{j\theta_2}$, then

$$z_1 z_2 = r_1 r_2 e^{j(\theta_1 + \theta_2)}.$$

The moduli are multiplied together, and the angles are added. Note that if $z = e^{j\theta}$, then $|z| = 1$ *for all values of* θ.

1.2 Trigonometric Interpolation

Suppose a function $f(\theta)$ is defined on the interval $(0, 2\pi)$, with f assumed to be periodic on the interval; thus, $f(\theta) = f(\theta \pm 2\pi)$.

Now consider constructing a trigonometric polynomial $p(\theta)$ to interpolate $f(\theta)$ of the form

$$p(\theta) = a_0 + \sum_{k=1}^{n} a_k \cos k\theta + b_k \sin k\theta. \tag{1.1}$$

This function has $2n+1$ coefficients, so it should be possible to interpolate f at $2n+1$ points. In the applications considered in this book, the points at which to interpolate are always *equally spaced* on the interval:

$$\theta_\ell = \frac{2\ell\pi}{2n+1}, \quad \ell = 0, 1, \ldots, 2n. \tag{1.2}$$

Let $x_\ell = f(\theta_\ell)$, and consider an example with $n = 2$. Then the interpolation conditions are $x_\ell = p(\theta_\ell)$, or

$$x_\ell = a_0 + a_1 \cos\theta_\ell + b_1 \sin\theta_\ell + a_2 \cos 2\theta_\ell + b_2 \sin 2\theta_\ell, \quad \ell = 0, 1, \ldots, 4.$$

This leads to the system of equations

$$\begin{bmatrix} 1 & \cos\theta_0 & \sin\theta_0 & \cos 2\theta_0 & \sin 2\theta_0 \\ 1 & \cos\theta_1 & \sin\theta_1 & \cos 2\theta_1 & \sin 2\theta_1 \\ 1 & \cos\theta_2 & \sin\theta_2 & \cos 2\theta_2 & \sin 2\theta_2 \\ 1 & \cos\theta_3 & \sin\theta_3 & \cos 2\theta_3 & \sin 2\theta_3 \\ 1 & \cos\theta_4 & \sin\theta_4 & \cos 2\theta_4 & \sin 2\theta_4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ b_1 \\ a_2 \\ b_2 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}.$$

Recall that $e^{j\theta} = \cos\theta + j\sin\theta$, which implies that

$$\cos\theta = \frac{e^{j\theta} + e^{-j\theta}}{2} \quad \text{and} \quad \sin\theta = \frac{e^{j\theta} - e^{-j\theta}}{2j}.$$

Using these in (1.1) with $n = 2$ yields

$$\begin{aligned} p(\theta) &= a_0 + \left(\frac{a_1}{2}\right) e^{j\theta} + \left(\frac{a_1}{2}\right) e^{-j\theta} + \left(\frac{b_1}{2j}\right) e^{j\theta} - \left(\frac{b_1}{2j}\right) e^{-j\theta} \\ &\quad + \left(\frac{a_2}{2}\right) e^{2j\theta} + \left(\frac{a_2}{2}\right) e^{-2j\theta} + \left(\frac{b_2}{2j}\right) e^{2j\theta} - \left(\frac{b_2}{2j}\right) e^{-2j\theta} \\ &= \left(\frac{a_2 + j\ b_2}{2}\right) e^{-2j\theta} + \left(\frac{a_1 + j\ b_1}{2}\right) e^{-j\theta} \\ &\quad + a_0 + \left(\frac{a_1 - j\ b_1}{2}\right) e^{j\theta} + \left(\frac{a_2 - j\ b_2}{2}\right) e^{2j\theta}. \end{aligned}$$

Giving the coefficients names corresponding to the powers of $e^{j\theta}$ yields

$$p(\theta) = X_{-2} e^{-2j\theta} + X_{-1} e^{-j\theta} + X_0 + X_1 e^{j\theta} + X_2 e^{2j\theta}. \tag{1.3}$$

Note that the coefficients appear in complex conjugate pairs. When the x_ℓ are real, it is straightforward to show that this is true in general. (See the next section.)

Recall (see (1.2)) that the points at which interpolation occurs are *evenly spaced*; that is, $\theta_\ell = \ell\theta_1$. Let $\omega = e^{j\theta_1} = e^{\frac{2j\pi}{2n+1}}$. Then all $e^{j\theta_\ell}$ can be expressed in terms of ω:

$$e^{j\theta_\ell} = e^{j\ell\theta_1} = \omega^\ell, \ell = 0, 1, \ldots, 2n.$$

Also, note that $\omega^{\ell} = \omega^{\ell \pm (2n+1)}$ and $\omega^{-\ell} = \omega^{-\ell \pm (2n+1)}$. For the example with $n = 2$, $\omega = e^{\frac{2j\pi}{5}}$, and the interpolation condition at θ_ℓ in (1.3) is

$$f(\theta_\ell) = x_\ell = p(\theta_\ell) = X_{-2}\omega^{-2\ell} + X_{-1}\omega^{-\ell} + X_0\omega^0 + X_1\omega^{\ell} + X_2\omega^{2\ell}.$$

Using the fact that $\omega^{-\ell} = \omega^{(2n+1-\ell)}$, and renaming the coefficients similarly ($X_{-\ell} \rightarrow X_{2n+1-\ell}$), the interpolation condition at x_ℓ becomes

$$x_\ell = X_0 + X_1\omega^{\ell} + X_2\omega^{2\ell} + X_3\omega^{3\ell} + X_4\omega^{4\ell},$$

which has to be satisfied for $\ell = 0, 1, \ldots, 4$:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \omega^4 \\ 1 & \omega^2 & \omega^4 & \omega^6 & \omega^8 \\ 1 & \omega^3 & \omega^6 & \omega^9 & \omega^{12} \\ 1 & \omega^4 & \omega^8 & \omega^{12} & \omega^{16} \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}. \tag{1.4}$$

This can be written as a matrix equation

$$\boldsymbol{MX} = \boldsymbol{x}.$$

It will be useful to have some additional properties of ω. First note that

$$1 + \omega + \omega^2 + \ldots + \omega^{2n} = 0.$$

This can be established by observing that the expression on the left side is a geometric sum equal to

$$\frac{1 - \omega^{2n+1}}{1 - \omega},$$

and this quantity is zero because $\omega^{2n+1} = 1$. For integers r and s one can show in a similar way that

$$\sum_{k=0}^{2n} \omega^{(kr-ks)} = \begin{cases} 0 & \text{if } r \neq s \\ 2n+1 & \text{if } r = s. \end{cases} \tag{1.5}$$

These simple results make solving $\boldsymbol{MX} = \boldsymbol{x}$ easy. To begin, let

$$\overline{\boldsymbol{M}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \bar{\omega} & \bar{\omega}^2 & \bar{\omega}^3 & \bar{\omega}^4 \\ 1 & \bar{\omega}^2 & \bar{\omega}^4 & \bar{\omega}^6 & \bar{\omega}^8 \\ 1 & \bar{\omega}^3 & \bar{\omega}^6 & \bar{\omega}^9 & \bar{\omega}^{12} \\ 1 & \bar{\omega}^4 & \bar{\omega}^8 & \bar{\omega}^{12} & \bar{\omega}^{16} \end{bmatrix}.$$

Then using (1.5) above, together with the fact that $\bar{\omega}^{\ell} = \omega^{-\ell}$, shows that $\overline{\boldsymbol{M}}\boldsymbol{M}$ is

$$\begin{bmatrix} 5 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 5 \end{bmatrix}.$$