FFT Library

Yuzhi Zhao

December 19, 2017

Contents

1 Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

2 Reference Material

This section records information for easy reference.

2.1 Table of Units

A trivial FFT library is unitless. FFT library can be implemented in different fields. The users of the library will ber responsible for knowing the meaning of input and giving proper unit.

2.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. The choice of symbols was made to be consistent with the heat transfer literature and with existing documentation for solar water heating systems. The symbols are listed in alphabetical order.

symbol	**unit**	**description**
a	none	real number term
b	none	coefficient of complex number term
ω_N	none	twiddle factor
$X(k)$	none	k^{th} term of DFT sequence
$X(n)$	none	n^{th} term of discrete sequence in time domain
$\mathcal{O}$	none	time complexity
$X_E(k)$	none	k^{th} term of DFT even term sequence
$X_O(k)$	none	k^{th} term of DFT odd term sequence
N	none	data size
r	none	radix number

2.3 Abbreviations and Acronyms

symbol	description
A	Assumption
DD	Data Definition
GS	Goal Statement
IM	Instance Model
SRS	Software Requirements Specification
FFT	Fast Fourier Transform
T	Theoretical Model
DFT	Discrete Fourier Transform
IDFT	Invers Discrete Transform

3 Introduction

Fast Fourier transforms are widely used in many applications in engineering, science, and mathematics fields. FFT's importance derives from the fact that in signal processing and image processing, it has made working in frequency domain equally computationally feasible as working in temporal or spatial domain.

The following section provides an overview of the Software Requirements Specification (SRS) for a FFT Library. The developed program will be referred to as Fast Fourier Transform Library (FFT). This section explains the purpose of this document, the scope of the system, the characteristics of the intended readers and the organization of the document.

3.1 Purpose of Document

This document will be used as a starting point for subsequent development phases, including writing the design specification **?** and the software verification and validation plan **?**. The design document will show how the requirements are to be realized, including decisions on the numerical algorithms and programming environment. The verification and validation plan will show the steps that will be used to increase confidence in the software documentation and the implementation.

3.2 Scope of the Family

The scope of the requirements includes specifying the input data requirements including the number of input data, the data type of input data, specifying the number of radix of FFT, which can also be understood as different calculation algorithms.

This FFT library will focus on radix 2 algorithm which is also known as the normal FFT algorithm, and radix 3 FFT algorithm.
Given the full information of input and parameters, the FFT Library will take advantage of FFT algorithms to do DFT or IDFT calculation effectively.

3.3 Characteristics of Intended Reader

Reviewers of this documentation should have a strong knowledge in Fourier Transform Theory. Also, reviewers should have an understanding of complex number calculation, as typically covered in first and second year Calulus courses.

3.4 Organization of Document

The organization of this document follows the template for an SRS for scientific computing software proposed by **?**. The presentation follows the standard pattern of presenting goals, theories, definitions, and assumptions. For readers that would like a more bottom up approach, they can start reading the instance models in Section 5.6 and trace back to find any additional information they require. The instance models provide the algebraic equations that model the FFT algorithm.

4 General System Description

This section identifies the interfaces between the system and its environment, describes the potential user characteristics and lists the potential system constraints.

4.1 Potential System Contexts

Figure 1 shows the system context. A circle represents an external entity outside the software, the user in this case. A rectangle represents the software system itself (FFT). Arrows are used to show the data flow between the system and its environment. Users could be any programs which need to use FFT Library.

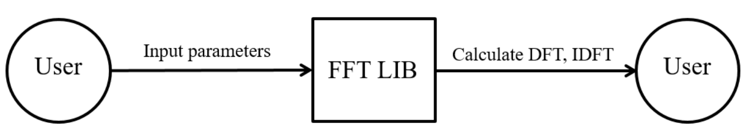

Figure 1: System Context

- User Responsibilities:
  - Provide the input data to the system, ensuring no errors in the data entry

  - Make sure invoking the right functions according to different input data type

- FFT Library Responsibilities:
  - Determine if the inputs satisfy the required data type and format
  - Calculate the required outputs
  - Detect data type mismatch because the input data should be numbers.

4.2 Potential User Characteristics

The end user of FFT Library should have an understanding of undergraduate Level 1 Calculus and Physics.

4.3 Potential System Constraints

There is no system constraints.

5 Commonalities

5.1 Background Overview

An FFT calculation can rapidly computes DFT by factorizing the DFT matrix into a product of sparse factors. As a result, it manages to reduce the complexity of computing the DFT from $\mathcal{O}(\log n^2)$, which arises if one simple applies the definition of DFT, to $\mathcal{O}(n \log n)$, where n is the data size. Because of this algorithm decreasing the amount of calculation incredibly so that the FFT is widely used in the digital signal processing, fast discrete Hartley transform.

5.2 Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning, with the purpose of reducing ambiguity and making it easier to correctly understand the requirements:

- Discrete Fourier Transform (DFT): A transform method to convert a signal from its original domain (often time or space) to a representation in the frequency domain.

- Inverse Disceret Fourier Transform (IDFT): A inverse transform method of DFT.

- Fast Fourier Transform (FFT): A fast algorithm computes the discrete Fourier transform (DFT) of a sequence.

- Inverse Fast Fourier Transform (IFFT): A fast algorithm computes the inverse discrete Fourier transform (IDFT) of a sequence.

5.3 Data Definitions

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given.

Number	DD1
Label	**Twiddle Factor**
Symbol	ω_N
SI Units	None
Equation	$\omega_N = e^{-2\pi i/N}$
Description	ω_N is twiddle Factor, a simpler term to represent $e^{-2\pi i/N}$. N is the data size. This representation simplifies the complex-structured equations, which can help readers focus on the transformation process.
Sources	
Ref. By	IM1, IM2

Number	DD2
Label	**Power Spectral Density (PSD)**
Symbol	$X(k)$
SI Units	None
Equation	$X(k) = X(0) + X(1) + X(2) + ..., (k \in \mathbb{Z}$, k is finite)
Description	Power spectral density (PSD) shows the strength of the variations (energy) as a function of frequency. In other words, it shows at which frequencies variations are strong and at which frequencies variations are weak.
Sources	http://www.cygres.com/OcnPageE/Glosry/SpecE.html
Ref. By	IM1, IM2, T1, T2

Number	DD3
Label	**Amplitude Function**
Symbol	$x(n)$
SI Units	None
Equation	Usually obtained from detector or other equipment directly.
Description	The amplitude of a periodic variable is a measure of its change over a single period (such as time or spatial period).
Sources	https://en.wikipedia.org/wiki/Amplitude
Ref. By	IM1, IM2, T1, T2

Number	DD4
Label	**The Odd Terms Of Power Spectral Density**
Symbol	$X(k)$
SI Units	None
Equation	$X_E(k) = X(1) + X(3) + ..., (k = 1, 3, 5...)$
Description	The odd term set of the complete PSD but keeping the same physical meaning as PSD.
Sources	
Ref. By	IM1

Number	DD5
Label	**The Even Terms Of Power Spectral Density**
Symbol	$X_O(k)$
SI Units	
Equation	$X_O(k) = X(0) + X(2) + ..., (k = 0, 2, 4...)$
Description	The even term set of the complete PSD but keeping the same physical meaning as PSD.
Sources	
Ref. By	IM1

5.4 Goal Statements

Given the input data array of $x(n)$ or $X(k)$, radix r, the goal statements are:

GS1: When given input, do FFT or IFFT calculation. The input could be real or complex numbers. FFT or IFFT calculation can use radix-2 or radix-3 algorithms.

5.5 Theoretical Models

This section focuses on the general equations and laws that FFT is based on.

Number	T1
Label	**Discrete Fourier Transform(DFT)**
Equation	$X(k) = \sum_{n=0}^{N-1} x(n)\ \omega_N$
Description	The above equation is the definition of Discrete Fourier Transform, which transforms a sequence of N complex numbers $x(0)$, $x(1)$, $x(2)$, $x(3)$... into another sequence of complex numbers, $X(0)$, $X(1)$, $X(2)$, $X(3)$, $X(4)$...In the equation, $x(n)$ is amplitude in time domain and ω_N is twiddle factor.
Source	http://dsp-book.narod.ru/TDCH/CH-02.PDF
Ref. By	IM1, IM2, T2

Number	T2
Label	**Inverse Discrete Fourier Transform(IDFT)**
Equation	$x(n) = \frac{1}{N} \sum_{n=0}^{N-1} X(k)\, \omega_N$
Description	The above equation is the definition of Inverse Discrete Fourier Transform, which transforms a sequence of N complex numbers $X(0)$, $X(1)$, $X(2)$, $X(3)$, $X(4)$... into another sequence of complex numbers, $x(0)$, $x(1)$, $x(2)$, $x(3)$...In the equation, $X(k)$ is amplitude in time domain and ω_N is twiddle factor in IDFT algorithm, while N is the length of sequence.
Source	https://en.wikipedia.org/wiki/Discrete_Fourier_transform
Ref. By	IM1, IM2

Number	T3
Label	**Euler's Formula**
Equation	$e^{ix} = cosx + isinx$
Description	Euler's formula, named after Leonhard Euler, is a mathematical formula in complex analysis that establishes the fundamental relationship between the trigonometric functions and the complex exponential function. e is the base of the natural logarithm, i is the imaginary unit, and cos and sin are the trigonometric functions cosine and sine respectively, with the argument x given in radians.
Source	https://en.wikipedia.org/wiki/Euler%27s_formula
Ref. By	IM1, IM2, T4

Number	T4
Label	**A Transform Of** ω_N^{kn}
Equation	$\omega_N^{akn} = \omega_{N/a}^{kn}$
Description	a is a coefficient. This transform is frequently used in FFT algorithms.
Source	
Ref. By	IM1, IM2

5.6 Instance Models

Number	IM1
Label	**Radix-2 FFT Calculation**
Input	a_n, b_n, N, r The input is constrained so that r = 2. If taking the real data as input, there is no b_n to be entered.
Output	X(k), k =(0, 1, ..., N), in format 'a + bi'
Description	X(k) is the power spectral density , a is the real number term, b is the coefficient part of complex number part. N is the size of data. r is the radix.
Sources	http://en.dsplib.org/content/fft_dec_in_time/fft_dec_in_time.html
Ref. By	-

Detailed Radix-2 FFT Algirithm

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n)\omega_N^{kn} \\ &= \sum_{even} x(n)\omega_N^{kn} + \sum_{odd} x(n)\omega_N^{kn} \end{aligned}$$

Define n = 2r and n = 2r + 1, r = 0, 1, 2, ..., N/2 -1.

$$
\begin{aligned}
X(k) &= \sum_{r=0}^{N/2-1} x(2r)\omega_N^{2kr} + \sum_{r=0}^{N/2-1} x(2r+1)\omega_N^{(2r+1)k} \\
&= \sum_{r=0}^{N/2-1} x(2r)(\omega_N^2)^{kr} + \omega_N^k \sum_{r=0}^{N/2-1} x(2r+1)(\omega_N^2)^{kr}
\end{aligned}
$$

According to Eular's Formula T3,

$$
\begin{aligned}
X(k) &= \sum_{r=0}^{N/2-1} x(2r)(\omega_N^2)^{kr} + \omega_N^k \sum_{r=0}^{N/2-1} x(2r+1)(\omega_N^2)^{kr} \\
&= \sum_{r=0}^{N/2-1} x(2r)(\omega_{N/2})^{kr} + \omega_N^k \sum_{r=0}^{N/2-1} x(2r+1)(\omega_{N/2})^{kr} \\
&= X_e(k) + \omega_N^k X_o(k)
\end{aligned}
$$

S

We can use FFT butterfly diagram to analysis a FFT computing process and we can clearly tell how the sequence is divided and computed. Here we will have a simplest N = 8, radix-2, butterfly diagram as example:

Figure 2

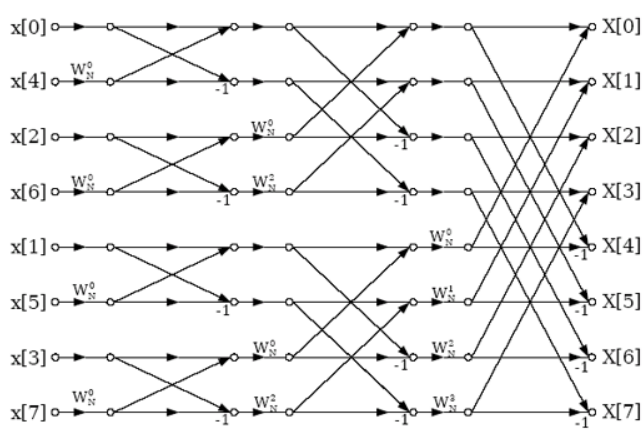

Figure 2: Radix-2 8-Point FFT

From the diagram, there were twice times of division of even and odd sequences. The first time division spilt the even and odd sequences. The second time, we took even and odd sequences retrieved from last division as two new sequences and split both of them into the even and odd sequences again. So far from the lowest level, we can consider our original sequence as a four-piece sequences. And then we can compute the sequence from the lowest level. Thus, we can develop our algorithm to a bigger domain. Unless the number of terms in a sequence is in pattern of $2^s (s \in \mathbb{Z}$, s < 20), the sequence can be reduced until the smallest piece is consist of two terms.

Radix-2 IFFT implements the same algorithim.

Number	IM2
Label	**Radix-3 FFT Calculation**
Input	a_n, b_n, N, r The input is constrained so that r = 3. If taking real data as input, there is no b_n to be entered.
Output	X(k), k =(0, 1, ..., N), in format 'a + bi'
Description	X(k) is the power spectral density , a is the real number part, b is the coefficient part of complex number part. N is the size of data. r is the radix.
Sources	
Ref. By	-

Detailed Radix-3 FFT Algirithm

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n)\omega_N^{kn} \\ &= \sum_{n=3^r} x(n)\omega_N^{kn} + \sum_{n=3^r+1} x(n)\omega_N^{kn} + \sum_{n=3^r+2} x(n)\omega_N^{kn} \\ &= \sum x_1(n)\omega_N^{kn} + \sum x_2(n)\omega_N^{kn} + \sum x_3(n)\omega_N^{kn} \end{aligned}$$

Define n = 3r, n = 3r + 1 and n = 3r +2, r = 0, 1, 2, ..., N/3 -1.

$$
\begin{aligned}
X(k) &= \sum_{r=0}^{N/3-1} x(3r)\omega_N^{3kr} + \sum_{r=0}^{N/3-1} x(3r+1)\omega_N^{(3r+1)k} + \sum_{r=0}^{N/3-1} x(3r+2)\omega_N^{(3r+2)k} \\
&= \sum_{r=0}^{N/3-1} x(3r)(\omega_N^{3kr}) + \omega_N^{k} \sum_{r=0}^{N/3-1} x(3r+1)(\omega_N^{3kr}) + \omega_N^{2k} \sum_{r=0}^{N/3-1} x(3r+2)(\omega_N^{3kr})
\end{aligned}
$$

According to Eular's Formula T3,

$$
\begin{aligned}
X(k) &= \sum_{r=0}^{N/3-1} x(3r)(\omega_N^{3kr}) + \omega_N^{k} \sum_{r=0}^{N/3-1} x(3r+1)(\omega_N^{3kr}) + \omega_N^{2k} \sum_{r=0}^{N/3-1} x(3r+2)(\omega_N^{3kr}) \\
&= \sum_{r=0}^{N/3-1} x(3r)(\omega_{N/3})^{kr} + \omega_N^{k} \sum_{r=0}^{N/3-1} x(3r+1)(\omega_{N/3})^{kr} + \omega_N^{2k} \sum_{r=0}^{N/3-1} x(3r+2)(\omega_{N/3})^{kr} \\
&= X_1(k) + \omega_N^{k} X_2(k) + \omega_N^{2k} X_3(k)
\end{aligned}
$$

We can use FFT butterfly diagram to analysis a FFT computing process and we can clearly tell how the sequence is divided and computed. Here we will have a simplest N = 9, radix-3, butterfly diagram as example:

Figure 3

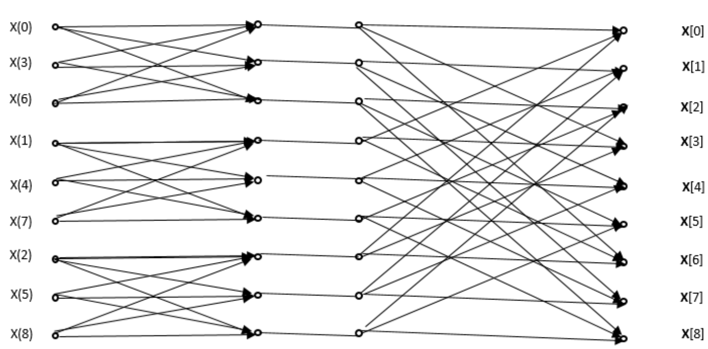

Figure 3: Radix-3 9-Point FFT

From the diagram, there were two times of division of sequences. Each division split sequence into three parts. The first sequence is the sum of all terms whose sequence number can be divided by 3 without the rest. The second sequence is the sum of all terms whose sequence number can be divided by 3 with rest of 1. The third sequence is the sum of all terms whose sequence number can be divided by 3 with the rest of 2. The first time division spilt the sequence into three parts following the rules as mentioned. The second time, we implement the same rule to split these three sequences so that each sequence is divided into three sequences. So far from the lowest level, we can consider our original sequence as a nine-piece sequences. And then we can compute the sequence from the lowest level. Thus, we can develop our algorithm to a bigger domain. Unless the number of terms in a sequence is in pattern of $3^s(s \in \mathbb{Z}, \text{s} < 20)$, the sequence can be reduced until the smallest piece is consist of three terms.

Radix-3 IFFT implements the same algorithim.

6 Variabilities

6.1 Assumptions

A1: The number of input of radix-2 FFT and IFFT shoud be $2^s(s \in \mathbb{Z}, \text{s} < 20)$ [IM1, DD4].

A2: The number of input of radix-3 FFT and IFFT should be $3^s(s \in \mathbb{Z}, \text{s} < 20)$ [IM2].

A3: The length of $X(k)$ sequence is always equal to the length of sequence $x(n)$ [T1, T2, IM1, IM2].

A4: The coefficient of every term in $x(n)$ is an integer for both complex and real number [IM1, IM2].

A5: When retrieving the complex number FFT function, some terms are still real numbers. Users enter 0 as an coefficient for complex terms under this occation [IM1, IM2].

A6: Assume the strides for $X(k)$ and $x(n)$ are both 1 [IM1, IM2, DD2, DD4, DD5].

A7: The length of input sequence is known [IM1, IM2].

6.2 Calculation

- Radix 2 FFT calculation algorithm [IM1]
- Radix 3 FFT calculation algortithm [IM2]

6.3 Output

Not Applicable.

7 Traceability Matrices and Graphs

The purpose of the traceability matrices is to provide easy references on what has to be additionally modified if a certain component is changed. Every time a component is changed, the items in the column of that component that are marked with an "X" should be modified as well. Table 1 shows the dependencies of theoretical models, data definitions, and instance models with each other. Table 2 shows the dependencies of theoretical models, data definitions, instance models, and likely changes on the assumptions.

	T1	T2	T3	T4	DD1	DD2	DD3	DD4	DD5	IM1	IM2
T1			X							X	X
T2										X	X
T3				X						X	X
T4										X	X
DD1										X	X
DD2	X	X								X	X
DD3	X	X								X	X
DD4										X	
DD5										X	
IM1											
IM2	X										

Table 1: Traceability Matrix Showing the Connections Between Items of Different Sections

The purpose of the traceability graphs is also to provide easy references on what has to be additionally modified if a certain component is changed. The arrows in the graphs represent dependencies. The component at the tail of an arrow is depended on by the component at the head of that arrow. Therefore, if a component is changed, the components that it points to should also be changed.

NOTE: Building a tool to automatically generate the graphical representation of the matrix by scanning the labels and reference can be future work.

	A1	A2	A3	A4	A5	A6	A7
T1			X				
T2			X				
T3							
T4							
DD1			X				
DD2						X	
DD3							
DD4	X						
DD5						X	
IM1	X		X	X	X	X	X
IM2		X	X	X	X	X	X

Table 2: Traceability Matrix Showing the Connections Between Assumptions and Other Items