CA presentation
A Fourier Series Library

Bo Cao, caob13@mcmaster.ca

September 30, 2019

Table of Contents I

Table of Contents II

Purpose of Document

An overview of a library for Fourier Series related computation.

- Introduce underlying mathematical theory.
- Introduce necessary adaptations of theory.
- Define a data structure storing Fourier Series data.
- Design and implement operations. (implementation not covered in the following slides)
- Review verification of library. (only a simple introduction shown)

Scope of the Family

- Limited length of Fourier Series - limited accuracy.
- Only accepts functions able to be converted to Fourier Series, no checking - hard to implement, leave to users.
- Binary operations require frequency agreements - better with explicit frequency conversion.
- Supports popular floating point types - but requires basic arithmetic operations and input/output.

Characteristics of Intended Reader

- Developers, maintainers and contributors.
- Potential users who want deep knowledge of this library (for compatibility analysis, code audition, etc.)

Organization of Document

Same as the template.

Potential System Contexts

User Responsibilities:

- Source code distribution, compatibility between compilers - compile by users.
- Check of input legality - library can detect some, but not all. Users have better understanding of library's used environments.

Library responsibilities:

- Report found input illegality - wrong format, etc.
- Perform operations in promised way.
- Get results within promised accuracy, analyze error when possible.
- Output within given format.
- Crash only when it is a must - zero memory leakage, no excessive CPU usage, etc.

Potential User Characteristics

Who uses this library?

- Scientific/industrial software developers - almost everyone have degrees/abundant knowledge in coding and math.
- Students preparing for these jobs - finished basic courses.

What do they know?

- Basic knowledge in Fourier Series - covered in advanced calculus courses, even real analysis
- Basic analysis of error estimation - taught in introductory scientific computation courses. (CAS 4X03, CAS 708/ CSE 700 as examples)
- Understanding of one of the languages in which this library provide APIs (C/C++, maybe Python, MATLAB, etc.)

These are the characteristics of potential users.

Potential System Constraints

- None until now.
- Possible suggestions welcomed.

Overview of Related Libraries

A list of related libraries

- FFTW: `www.fftw.org`;
- Java Fourier Transform Library: `https://github.com/tambapps/fourier-transform-library`, written in Java;
- FINUFFT: `https://finufft.readthedocs.io/en/latest/`;
- Fourier Transform in MATLAB: `https://www.mathworks.com/help/symbolic/fourier.html`, possible operations in other packages.

Analysis:

- Most written in C/C++;
- Most only implement transformation;
- Fast transformation.

What do we focus on?

Requirements on our library:

- Focus on operation;
- Might need to implement conversion to/from them.

Terminology and Definitions

Fourier Series:

$$f(t) = A_0 + \sum_{i=1}^{+\infty} A_i \cos(i\omega t) + \sum_{i=1}^{+\infty} B_i \sin(i\omega t). \tag{1}$$

- $f(t)$: transformed function;
- ω: base frequency;
- A_i, B_i: Fourier Series.

Data Definitions

What is enough to describe a Fourier Series?

- ω;
- A_i, B_i.

Can our library handle infinite series? No.
What to do? Cut off!
Where? Let users decide.
Additional information: cut-off position n, only $A_i, i = 0 : n$ and $B_i, i = 1 : n$ are computed, stored and outputted.
Will this library estimate error? Maybe. See if it is easy to implement.

Goal Statements: Quoted from Problem Statements

I intend to implement a library for Fourier series related computation. I will mainly implement the following functions.

- Compute the Fourier series of a given function.
- Compute the value of a function with given variable value and the Fourier series of this function.
- Implement the addition, subtraction, multiplication and division of Fourier series. That is, suppose that the functions $f(t)$ and function $g(t)$ have Fourier series F and G, respectively, this library computes the Fourier series of $f(t) + g(t)$, $f(t) - g(t)$, $f(t) * g(t)$, and $f(t)/g(t)$ from F and G.
- Implement some basic functions (sin, exp, etc.) of Fourier series. That is, suppose a function $f(t)$ with known Fourier series F, and a known basic function $g()$, we would like to compute the Fourier series of $g(f(t))$ from F.
- Formatted input and output of Fourier series.

Possible Extensions

- Comparison between Fourier series;
- Amplitude of Fourier Series;
- Base frequency reduction ($\omega \to (1/k)\omega,\ k \in \mathbb{Z}^+$);
- Conversion from/to other data formats.

Theoretical Models

- Conversion from functions: Fast Fourier Transform (FFT).
- Computation of function values: direct and easy.
- Addition, subtraction, multiplication: use the operations of two Fourier Series to find relationships between coefficients.
- Division: a little hard, covered later.
- Functions of Fourier Series: convert functions to Taylor Series, replace independent variables with Fourier Series, find relationships between coefficients.
- Input/output: direct and easy.

How to define and compute division?

Two ways for computing the Fourier Series of $f(t)/g(t)$ from $f(t)$'s Fourier Series $A_{f,i}, B_{f,i}$ and Fourier Series $A_{g,i}, B_{g,i}$.
Define the solution by $A_{d,i}, B_{d,i}$.
Ways:

Way 1. Use $[f(t)/g(t)] * g(t) = f(t)$, build linear equations among $A_{f,i}, B_{f,i}$, $A_{g,i}, B_{g,i}$, $A_{d,i}, B_{d,i}$. Solve equations for $A_{d,i}, B_{d,i}$.

Way 2. Calculate Fourier Series of $1/g(t)$ like Way 1, and multiply it with that of $f(t)$.

Same? Not with cut-off errors, but can be proven equal within this error.
Both throws exceptions when unique solution cannot be found for linear equations. (Same as division by zero in normal divisions?)
Advantage compared to others.

Way 1. Intuitive.

Way 2. Reusability for multiple $f(t)$'s divided by one $g(t)$.

Have not decided which one to use. Any suggestions?

Variability in Assumptions

- How long are the series before cut-off?
- Data type to store coefficients and base frequency.
- How to get the input? Conversion from what other data formats?

Variability in Calculation

- Fourier transform part: one algorithm, multiple ones chosen by user, or chosen by library?
- What linear solver to use?
- Cut-off method: just cut-off or let cut-off terms slightly modify remaining terms?
- Cut-off in Taylor Series. How long? Relationship with Fourier Series cut-off? Who makes the decision?

Variability in Output

- Output/conversion format. Which ones to implement?
- Accuracy in formatted output.
- Too large to be outputted? Maybe not.

Functional Requirements on Functions Converted to Fourier Series

Quoted from the Problem Statement:
The aforementioned function shall be a $\mathbb{R} \to \mathbb{R}$ function, whose Fourier series exists. Instead of verification by this library (due to foreseeable technical difficulties), this property shall be guaranteed by the users.

Functional Requirements: Others

- Size: not too large. (usually)
- Overflow? Users shall prevent, library just throws exception.

Non-functional requirements

- Implement easy algorithms first, no much need on speed - low data size.
- Compiler/dependent library compatibility - copy/implement solvers or call them?

Verification

- Verify with simple functions.
- Error estimation needed for verification? Maybe implement a simple but relaxed error estimation.
- How many kinds of input? All $f(t)$ and $g(t)$ cover as many types of functions as possible.

To be expanded in the future.

Traceability

TBD.