

Fourier analysis

BY YOUJUN HU

Institute of Plasma Physics, Chinese Academy of Sciences

Email: yjhu@ipp.cas.cn

Abstract

This note discusses the Discrete Fourier Transform (DFT) and its variations (e.g., the discrete sine transform).

1 Introduction

This note discusses the Fourier expansion and Discrete Fourier Transform (DFT), giving step by step derivation of DFT and its variations (e.g., the discrete sine transform). After carefully going through these elementary derivations, I feel more comfortable when using DFT in my numerical codes.

(The Fast Fourier Transform algorithm (FFT) makes the DFT fast enough to solve many real-life problems, which makes FFT be among the top algorithms that have changed the world. The FFT algorithm remained mysterious to me for many years until I read Cooley and Tukey's original paper, which turns out to be a very concise paper and easy to follow (details are discussed in [Append. A](#)).

2 Fourier series

2.1 Use trigonometric basis functions

If $h(x)$ is a function of period $2L$, then it can be proved that $h(x)$ can be expressed as the following series

$$h(x) = \sum_{n=0}^{\infty} a_n \cos\left(\frac{n\pi}{L}x\right) + \sum_{n=1}^{\infty} b_n \sin\left(\frac{n\pi}{L}x\right), \quad (1)$$

which is called Fourier series. It is not trivial to prove the above statement (what is needed in the proof is to prove that the set of functions $\cos(n\pi x/L)$ and $\sin(n\pi x/L)$ with $n = 0, 1, \dots, \infty$ is a “complete set”[1]). I will ignore this proof and simply start with Eq. (1).

At this point it is not clear what the coefficients a_n and b_n are. These can be obtained by taking product of Eq. (1) with $\cos(j\pi x / L)$ and $\sin(j\pi x / L)$, respectively, and then integrating from $-L$ to L , which gives

$$a_0 = \frac{1}{2L} \int_{-L}^L h(x) dx, \quad (2)$$

and, for $j \geq 1$,

$$a_j = \frac{1}{L} \int_{-L}^L h(x) \cos\left(\frac{j\pi}{L}x\right) dx, \quad (3)$$

$$b_j = \frac{1}{L} \int_{-L}^L h(x) \sin\left(\frac{j\pi}{L}x\right) dx. \quad (4)$$

In order to enable a_n to be uniformly expressed by Eq. (3), Fourier series are often redefined as

$$h(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left[a_n \cos\left(\frac{n\pi}{L}x\right) + b_n \sin\left(\frac{n\pi}{L}x\right) \right]. \quad (5)$$

[Fourier series (5) can also be written as

$$h(x) = \sum_{n=-\infty}^{\infty} \left[\frac{a_n}{2} \cos\left(\frac{n\pi}{L}x\right) + \frac{b_n}{2} \sin\left(\frac{n\pi}{L}x\right) \right], \quad (6)$$

where n ranges from $-\infty$ to ∞ , and there is no special treatment for the edge case of $n = 0$. In obtaining expression (6) from (5), use has been made of $a_n \cos(n\pi x / L) = a_{-n} \cos(-n\pi x / L)$ and $b_n \sin(n\pi x / L) = b_{-n} \sin(-n\pi x / L)$.]

Note that sine and cosine are similar to each other: one can be obtained from the other by shifting, i.e., they differ only in the “phase”. Using trigonometric identities, expression (5) can be expressed in terms of only cosine:

$$h(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} A_n \cos\left(\frac{n\pi}{L}x - \Phi_n\right), \quad (7)$$

where the amplitude A_n is given by

$$A_n = \sqrt{a_n^2 + b_n^2}, \quad (8)$$

and the phase Φ_n is given by

$$\Phi_n = \text{atan2}(b_n, a_n), \quad (9)$$

where $\text{atan2}(y, x)$ is the 2-argument arctangent, which gives an angle in the correct quadrant. Expression (7) can be stated as: a periodic signal is composed of cosine functions of different frequencies and phases (we interpret x as time).

If $h(x)$ is a complex-valued function (the independent variable x is still real number), then the above Fourier expansion can be applied to its real part and imaginary part, respectively. Combining the results, we can see that Eqs. (3)-(5) is still valid. In this case, a_n and b_n are complex numbers.

2.2 Use basis functions $e^{in\pi x/L}$

Fourier series are often expressed in terms of the complex-valued basis functions $e^{in\pi x/L}$. Next, we derive this version of the Fourier series, which is the most popular version we see in textbooks and papers (we will see why this version is popular).

Using Euler's formula (this is a bridge between representations using real numbers and complex numbers)

$$\cos\left(\frac{n\pi}{L}x\right) = \frac{e^{in\pi x/L} + e^{-in\pi x/L}}{2} \quad (10)$$

and

$$\sin\left(\frac{n\pi}{L}x\right) = \frac{e^{in\pi x/L} - e^{-in\pi x/L}}{2i} \quad (11)$$

in Eq. (5), we obtain

$$h(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left[\left(\frac{a_n - ib_n}{2} \right) e^{in\pi x/L} + \left(\frac{a_n + ib_n}{2} \right) e^{-in\pi x/L} \right]. \quad (12)$$

Define

$$c_n = \frac{a_n - ib_n}{2}, \quad (13)$$

then, by using $a_{-n} = a_n$ $b_{-n} = -b_n$, we know that

$$c_{-n} = \frac{a_n + ib_n}{2}. \quad (14)$$

Then Eq. (12) is written

$$h(x) = \sum_{n=-\infty}^{\infty} c_n e^{in\pi x/L}. \quad (15)$$

Using the expressions of a_n and b_n given by Eq. (3) and (4), c_n is expressed as

$$c_n = \frac{1}{2L} \int_{-L}^L h(x) e^{-in\pi x/L} dx. \quad (16)$$

Equation (15) along with Eq. (16) is the version of Fourier series using complex basis functions. In this version, the index n is an integer ranging from $-\infty$ to $+\infty$, which is unlike Eq. (1), where n is from 0 to $+\infty$. An advantage of Eqs. (15) and (16) is that no special treatment is needed for the edge case of $n=0$.

Let's recover the trigonometric version from the complex version. Using Eq. (16) in Eqs. (15), we obtain

$$\begin{aligned}
 h(x) &= \sum_{n=-\infty}^{\infty} \left(\frac{1}{2L} \int_{-L}^L h(x) e^{-in\pi x/L} dx \right) e^{in\pi x/L} \\
 &= \sum_{n=-\infty}^{\infty} \left(\frac{1}{2L} \int_{-L}^L h(x) \left(\cos\left(\frac{n\pi}{L}x\right) - i \sin\left(\frac{n\pi}{L}x\right) \right) dx \right) \left(\cos\left(\frac{n\pi}{L}x\right) + i \sin\left(\frac{n\pi}{L}x\right) \right) \\
 &= \sum_{n=-\infty}^{\infty} \left(\frac{1}{2L} \int_{-L}^L h(x) \cos\left(\frac{n\pi}{L}x\right) dx \right) \cos\left(\frac{n\pi}{L}x\right) \\
 &\quad + \sum_{n=-\infty}^{\infty} \left(\frac{1}{2L} \int_{-L}^L h(x) \sin\left(\frac{n\pi}{L}x\right) dx \right) \sin\left(\frac{n\pi}{L}x\right) \\
 &\quad + \sum_{n=-\infty}^{\infty} \left(\frac{1}{2L} \int_{-L}^L h(x) \cos\left(\frac{n\pi}{L}x\right) dx \right) \left(i \sin\left(\frac{n\pi}{L}x\right) \right) \\
 &\quad + \sum_{n=-\infty}^{\infty} \left(\frac{1}{2L} \int_{-L}^L h(x) \left(-i \sin\left(\frac{n\pi}{L}x\right) \right) dx \right) \left(\cos\left(\frac{n\pi}{L}x\right) \right)
 \end{aligned} \tag{17}$$

$$\begin{aligned}
 &\quad + \sum_{n=-\infty}^{\infty} \left(\frac{1}{2L} \int_{-L}^L h(x) \left(-i \sin\left(\frac{n\pi}{L}x\right) \right) dx \right) \left(\cos\left(\frac{n\pi}{L}x\right) \right)
 \end{aligned} \tag{18}$$

Noting that the $+n$ terms cancel the $-n$ terms in both line (17) and line (18), the above expression is reduced to

$$\begin{aligned}
 h(x) &= \sum_{n=-\infty}^{\infty} \left(\frac{1}{2L} \int_{-L}^L h(x) \cos\left(\frac{n\pi}{L}x\right) dx \right) \cos\left(\frac{n\pi}{L}x\right) \\
 &\quad + \sum_{n=-\infty}^{\infty} \left(\frac{1}{2L} \int_{-L}^L h(x) \sin\left(\frac{n\pi}{L}x\right) dx \right) \sin\left(\frac{n\pi}{L}x\right)
 \end{aligned} \tag{19}$$

Define new expansion coefficients

$$a'_n = \frac{a_n}{2} = \frac{1}{2L} \int_{-L}^L h(x) \cos\left(\frac{n\pi}{L}x\right) dx \tag{20}$$

and

$$b'_n = \frac{b_n}{2} = \frac{1}{2L} \int_{-L}^L h(x) \sin\left(\frac{n\pi}{L}x\right) dx \tag{21}$$

then expression (19) is written as

$$h(x) = \sum_{n=-\infty}^{\infty} a'_n \cos\left(\frac{n\pi}{L}x\right) + \sum_{n=-\infty}^{\infty} b'_n \sin\left(\frac{n\pi}{L}x\right), \tag{22}$$

which recovers expression (6). Note that $n \in (-\infty, +\infty)$ in this version, which differs from Eq. (5). Also this expression has no edge case that needs special treatment.

As a benchmark, we can start from expression (6) to derive the complex Fourier expansion: Using the Euler formula in expression (6), we obtain

$$\begin{aligned}
 h(x) &= \sum_{n=-\infty}^{\infty} \left[\left(\frac{a_n}{4} - \frac{ib_n}{4} \right) e^{in\pi x/L} + \left(\frac{a_n}{4} + \frac{ib_n}{4} \right) e^{-in\pi x/L} \right] \\
 &= \sum_{n=1}^{\infty} \left(\frac{a_n}{4} - \frac{ib_n}{4} \right) e^{in\pi x/L} + \sum_{n=-1}^{-\infty} \left(\frac{a_n}{4} + \frac{ib_n}{4} \right) e^{-in\pi x/L} \\
 &\quad + \sum_{n=-\infty}^0 \left(\frac{a_n}{4} - \frac{ib_n}{4} \right) e^{in\pi x/L} + \sum_{n=\infty}^0 \left(\frac{a_n}{4} + \frac{ib_n}{4} \right) e^{-in\pi x/L}
 \end{aligned}$$

Noting that $a_n = a_{-n}$ and $b_n = -b_{-n}$, the above expression is written as

$$h(x) = \sum_{n=-\infty}^{\infty} \left(\frac{a_n}{2} - \frac{ib_n}{2} \right) e^{in\pi x/L}.$$

Using Eq. (13), the coefficients a_n and b_n appearing in Eq. (5) can be recovered from c_n by

$$a_n = c_n + c_{-n}, \quad (23)$$

$$b_n = i(c_n - c_{-n}). \quad (24)$$

If $h(x)$ is real, then the coefficients a_n and b_n are real. Then equation (13) implies that c_n and c_{-n} are complex conjugates. In this case, expressions (23) and (24) are simplified as

$$a_n = 2\text{Re}(c_n), \quad (25)$$

$$b_n = -2\text{Im}(c_n). \quad (26)$$

[In the above, we use the basis functions $e^{in\pi x/L}$ to expand $h(x)$. If we choose the basis functions to be $e^{-in\pi x/L}$, then it is ready to verify that the Fourier series are written

$$h(x) = \sum_{n=-\infty}^{\infty} c_n e^{-in\pi x/L}, \quad (27)$$

with c_n given by

$$c_n = \frac{1}{2L} \int_{-L}^L h(x) e^{in\pi x/L} dx. \quad (28)$$

In this case, the coefficients a_n and b_n can be recovered from c_n by

$$a_n = c_n + c_{-n} \quad (29)$$

$$b_n = -i(c_n - c_{-n}) \quad (30)$$

In using the Fourier series, we should be aware of which basis functions are used.]

2.3 Numerical evaluation of Fourier expansion coefficients

For a periodic function $h(t)$ with a period of T , its Fourier expansion is given by

$$h(t) = \sum_{n=-\infty}^{\infty} c_n \exp\left(in \frac{2\pi}{T} t\right), \quad (31)$$

with the expansion coefficient c_n given by

$$c_n = \frac{1}{T} \int_0^T h(t) \exp\left(-in \frac{2\pi}{T} t\right) dt. \quad (32)$$

How to numerically compute c_n ? A simple way is to use the rectangle formula to approximate the integration in Eq. (32), i.e.,

$$c_n \approx \frac{\Delta}{T} \sum_{j=0}^{N-1} h_j \exp\left(-in \frac{2\pi j \Delta}{T}\right), \quad (33)$$

where $h_j = h(t_j)$ and $t_j = j\Delta$ with $j = 0, 1, 2, \dots, N-1$ and $\Delta = T/N$, as is shown in Fig. 1.

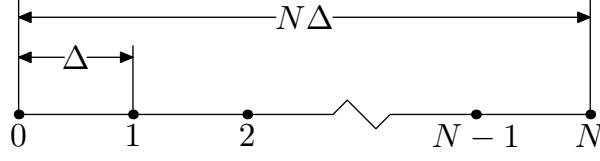


Figure 1. Sampling points in one period of the signal, where $T = N\Delta$ is the period of the signal.

Note Eq. (33) is an approximation, which will become exact if $\Delta \rightarrow 0$. In practice, we can sample $h(t)$ only with a nonzero Δ . Therefore Eq. (33) is usually an approximation. Do we have some rules to choose a suitable Δ so that Eq. (33) can become a good approximation or even an exact relation? This important question is answered by the sampling theorem (will be discussed in Append. B), which states that a suitable Δ to make Eq. (33) exact is given by $\Delta \leq 1/(2f_c)$, where $f_c = N_c/T$ is the largest frequency contained in $h(t)$ (i.e., c_n is zero for $|n| \geq N_c$).

$f_s \equiv 1/\Delta$ is called sampling frequency. Then the above condition can be rephrased as: the sampling frequency should be larger than $2f_c$.

3 Discrete Fourier transformation (DFT)

3.1 Definition

Denote the summation in Eq. (33) by H_n :

$$H_n \equiv \sum_{j=0}^{N-1} h_j \exp\left(-i2\pi \frac{nj}{N}\right). \quad (34)$$

Then H_n with $n = 0, 1, \dots, N-1$ is called the **Discrete Fourier transformation (DFT)** of h_j with $j = 0, 1, \dots, N-1$. (The efficient algorithm (FFT) of computing the DFT is discussed in Appendix A.)

Why do we choose n to be in the positive range $[0: N-1]$? This is to make the transform pair symmetric: both j and n are in the same range. This symmetry makes the transform pair easy to remember. This does not mean we only need the positive frequency components. In fact, we usually need n in the range $[-N/2, N/2]$. This will be discussed in the next section.

In terms of H_n , the Fourier expansion coefficients c_n in Eq. (33) is written as

$$c_n = \frac{1}{N} H_n. \quad (35)$$

The corresponding frequency is

$$f_n = \frac{n}{T} = \frac{n}{\Delta N} = f_s \frac{n}{N}. \quad (36)$$

Here $f_1 = 1/T$ is called the fundamental frequency (spacing in the frequency domain, i.e., frequency resolution).

3.2 Periodic property

We are usually interested in Fourier expansion coefficients in the range $n \in \left[-\frac{N}{2}, \frac{N}{2}\right]$ (assume that N is even) because we expect the coefficients decay with $|n|$ increasing (so we impose a cutoff for a range that is symmetric about $n = 0$). However, the original DFT is for the range $n \in [0: N - 1]$. How do we reconcile them? The answer is that we make use of the periodic property of DFT. It is obvious that H_n defined in Eq. (34) has the following periodic property:

$$H_{n+N} = H_n. \quad (37)$$

Using this, we can infer what we need: the value of each H_n with $n \in \left[-\frac{N}{2}, -1\right]$ is equal to H_n with $n \in \left[\frac{N}{2}, N - 1\right]$, element by element. Then we can associate H_j for $j \in [0: N - 1]$ with frequency f_j defined by:

$$f_j = \begin{cases} j f_1, & \text{for } 0 \leq j \leq \frac{N}{2} \\ (j - N) f_1, & \text{for } \frac{N}{2} + 1 \leq j \leq N - 1 \end{cases}. \quad (38)$$

3.3 Frequency resolution and bandwidth

The frequency interval between neighbor DFT points is $1/T$, where T is the time-window in which the signal is sampled. This frequency interval is called frequency resolution (resolution in the frequency domain), which is determined only by the length of the time-window and is independent of the sampling frequency. If the time-window is fixed, increasing the sampling frequency only increase the frequency range of the DFT ($[-f_s/2: f_s/2]$), which is called bandwidth, and the frequency interval between neighbor DFT points are still $1/T$, i.e., the frequency resolution is not changed.

4 Inverse transform

4.1 Reconstruct the original function using DFT

The Fourier series of $h(t)$

$$h(t) = \sum_{n=-\infty}^{\infty} c_n \exp\left(-i \frac{n 2\pi t}{T}\right), \quad (39)$$

can be approximated as

$$h(t) \approx \sum_{n=-N/2}^{N/2} c_n \exp\left(-i \frac{n 2\pi t}{T}\right). \quad (40)$$

Using the relation $c_n = H_n/N$, the above equation is written as

$$\begin{aligned} h(t) &= \frac{1}{N} \sum_{n=-N/2}^{N/2} H_n \exp\left(-i \frac{n2\pi t}{T}\right) \\ &= \frac{1}{N} \left[\sum_{n=0}^{N/2} H_n \exp\left(-i \frac{n2\pi t}{T}\right) + \sum_{n=-N/2}^{-1} H_n \exp\left(-i \frac{n2\pi t}{T}\right) \right]. \end{aligned} \quad (41)$$

Using the periodic property of DFT, i.e., $H_n = H_{N+n}$, the above expression is written as

$$h(t) = \frac{1}{N} \left[\sum_{n=0}^{N/2} H_n \exp\left(-i \frac{n2\pi t}{T}\right) + \sum_{n=N/2}^{N-1} H_n \exp\left(-i \frac{(n-N)2\pi t}{T}\right) \right]. \quad (42)$$

Equation (42) provides a formula of constructing an approximate function using the DFT of the discrete samplings of the original function.

4.2 Evaluate the reconstructed function at discrete points

Evaluate $h(t)$ given by Eq. (42) at the discrete point $t = j\Delta$, yielding

$$\begin{aligned} h(j\Delta) &= \frac{1}{N} \left[\sum_{n=0}^{N/2} H_n \exp\left(-i \frac{n2\pi j}{N}\right) + \sum_{n=N/2}^{N-1} H_n \exp\left(-i \frac{(n-N)2\pi j}{N}\right) \right] \\ &= \frac{1}{N} \left[\sum_{n=0}^{N/2} H_n \exp\left(-i \frac{n2\pi j}{N}\right) + \sum_{n=N/2}^{N-1} H_n \exp\left(-i \frac{n2\pi j}{N}\right) \right] \\ &= \frac{1}{N} \left[\sum_{n=0}^{N-1} H_n \exp\left(-i \frac{n2\pi j}{N}\right) + H_{N/2} \exp(-i\pi j) \right]. \end{aligned} \quad (43)$$

Drop the blue term (which is negligible if h satisfies the condition required by the sampling theorem), then $h(j\Delta)$ is written as

$$h(j\Delta) \approx \frac{1}{N} \sum_{n=0}^{N-1} H_n \exp\left(-i \frac{n2\pi j}{N}\right). \quad (44)$$

The right-hand side of Eq. (44) turns out to be the inverse DFT discussed in Sec. 4.3.

4.3 Inverse DFT

The DFT in Eq. (34), i.e.,

$$H_n \equiv \sum_{j=0}^{N-1} h_j \exp\left(i \frac{2\pi}{N} n j\right), \quad (45)$$

with $j = 0, 1, 2, \dots, N - 1$ and $n = 0, 1, 2, \dots, N - 1$ can also be considered as a set of linear algebraic equations for h_j and can be solved in terms of h_j , which gives

$$h_j = \frac{1}{N} \sum_{n=0}^{N-1} H_n \exp\left(-i \frac{2\pi}{N} n j\right). \quad (46)$$

(The details on how to solve Eq. (34) to obtain the solution (46) is provided in Sec. 4.4.) Equation (46) recovers h_j from H_n (i.e., the DFT of h_j), and thus is called the inverse DFT.

The normalization factor multiplying the DFT and inverse DFT (here 1 and $1/N$) and the signs of the exponents are merely conventions. The only requirement is that the DFT and inverse DFT have opposite-sign exponents and that the product of their normalization factors be $1/N$. In most FFT computer libraries, the $1/N$ factor is omitted. So one must manually include this factor when one wants to reproduce the original data by a forward and then a backward transform.

4.4 Proof of the inverse DFT

In order to solve the linear algebraic equations (34) for h_j , multiply both sides of each equation by $\exp(-i \frac{2\pi}{N} n J)$, where J is an integer between $[0, N - 1]$, and then add all the equations together, which yields

$$\sum_{n=0}^{N-1} \exp\left(-i \frac{2\pi}{N} n J\right) H_n = \sum_{n=0}^{N-1} \sum_{j=0}^{N-1} h_j \exp\left(i \frac{2\pi}{N} n(j - J)\right). \quad (47)$$

Interchanging the sequence of the two summation on the right-hand side, equation (47) is written

$$\sum_{n=0}^{N-1} \exp\left(-i \frac{2\pi}{N} n J\right) H_n = \sum_{j=0}^{N-1} h_j \sum_{n=0}^{N-1} \exp\left(i \frac{2\pi}{N} n(j - J)\right). \quad (48)$$

Using the fact that (verified by Wolfram Mathematica)

$$\sum_{n=0}^{N-1} \exp\left[i \frac{2\pi}{N} n(j - J)\right] = N \delta_{jJ}, \quad (49)$$

where δ_{jJ} is the Kroneker Delta, equation (48) is written

$$\sum_{n=0}^{N-1} \exp\left(-i \frac{2\pi}{N} n J\right) H_n = \sum_{j=0}^{N-1} h_j N \delta_{jJ}, \quad (50)$$

i.e.,

$$\sum_{n=0}^{N-1} \exp\left(-i \frac{2\pi}{N} n J\right) H_n = N h_J, \quad (51)$$

which can be solved to give

$$h_J = \frac{1}{N} \sum_{n=0}^{N-1} \exp\left(-i \frac{2\pi}{N} n J\right) H_n. \quad (52)$$

Equation (52) is the inverse DFT.

5 About using the FFT library

FFTW/scipy uses negative exponents as the forward transform and the positive exponents as inverse transform. Specifically, the forward DFT in FFTW is defined by

$$H_n \equiv \sum_{j=0}^{N-1} h_j \exp\left(-i \frac{2\pi}{N} n j\right), \quad (53)$$

and the inverse DFT is defined by

$$h_j = \sum_{n=0}^{N-1} H_n \exp\left(i \frac{2\pi}{N} n j\right), \quad (54)$$

where there is no $1/N$ factor in the inverse DFT, and thus this factor should be included manually if we want to recover the original data from the inverse DFT. (In some rare cases, e.g. in the book “Numerical recipe”[2], positive exponents are used in defining the forward Fourier transformation and negative exponents are used in defining the backward one. When using a Fourier transformation library, it is necessary to know which convention is used in order to correctly use the output of the library.)

In Scipy:

```
def fft(x, n=None, axis=-1, norm=None, overwrite_x=False,
workers=None, *,
      plan=None):
    #norm : {"backward", "ortho", "forward"}, optional
    #       Normalization mode. Default is "backward", meaning no
    #       normalization on
    #       the forward transforms and scaling by 1/n on the ifft.
    #       "forward" instead applies the 1/n factor on the forward
    #       transform.
    #       For norm="ortho", both directions are scaled by 1/
    #       sqrt(n).
```

I use the Fortran interface of the FFTW library. To have access to FFTW library, use the following codes:

```
use, intrinsic :: iso_c_binding
implicit none
```

```
include 'fftw3.f03'
```

Here the first line uses `iso_c_binding` module to interface with C in which FFTW is written. To use the FFT subroutines in FFTW, we need to define some variables of the desired types, such as

```
type(C_PTR) :: plan1, plan2
complex(C_DOUBLE_COMPLEX) :: in(0:n-1), out(0:n-1)
```

Specify what kind of transform to be performed by calling the corresponding “planner” routine:

```
plan1 = fftw_plan_dft_1d(n, in,out, FFTW_FORWARD,FFTW_ESTIMATE)
```

Here the “planner” routine for one-dimensional DFT is called. One thing that the “planner” routine does is to factor the matrix M_{kj} mentioned above, in order to get prepared for performing the actual transform. Therefore “planner” do not need the actual data stored in “in” array. What is needed is the length and numerical type of “in” array. It is obvious that the “planner” routine needs to be invoked for only once for a given type of array with the same length.

Store input data in the “in” arrays, then, we can perform a DFT by the following codes:

```
call fftw_execute_dft(plan1, in, out)
```

Similarly, we can perform a inverse DFT by the following codes:

```
plan2 = fftw_plan_dft_1d(ngrids, in,out,FFTW_BACKWARD,
FFTW_ESTIMATE)
call fftw_execute_dft(plan2, in, out)
```

After all the transforms are done, we need to manually de-allocate the arrays created by the “planner” routine by calling the “fftw_destroy_plan” routine:

```
call fftw_destroy_plan(plan2)
```

(Unless they are local arrays, Fortran does not automatically de-allocate arrays allocated by the `allocate()`, so manually de-allocate all allocated arrays is necessary for avoid memory leak)

6 Discrete sine transform and cosine transform

We mentioned (without giving proof) that the set of functions $\cos(n2\pi(x - x_0)/(2L))$ and $\sin(n2\pi(x - x_0)/(2L))$ with $n = 0, 1, \dots, \infty$ is a “complete set” in expanding any function in the interval $(x_0, x_0 + 2L)$, where x_0 is an arbitrary

point. Therefore Fourier series use both cosine and sine as basis functions to expand a function. Let us introduce another conclusion (again without giving proof) that the set of sine functions $\sin(n\pi(x - x_0)/(2L))$ with $n = 1, 2, \dots, \infty$ is a “complete set” in expanding any function h in the interval $(x_0, x_0 + 2L)$. A similar conclusion is that the set of cosine functions $\cos(n\pi(x - x_0)/(2L))$ with $n = 0, 1, 2, \dots, \infty$ is a “complete set” in expanding any function h in the interval $(x_0, x_0 + 2L)$. Note that the argument of the basis functions used in the Fourier expansion differ from those used in the sine/cosine expansion by a factor of two.

The first five basis functions used in Fourier expansion, sine expansion, and cosine expansion are plotted in Fig. 2.

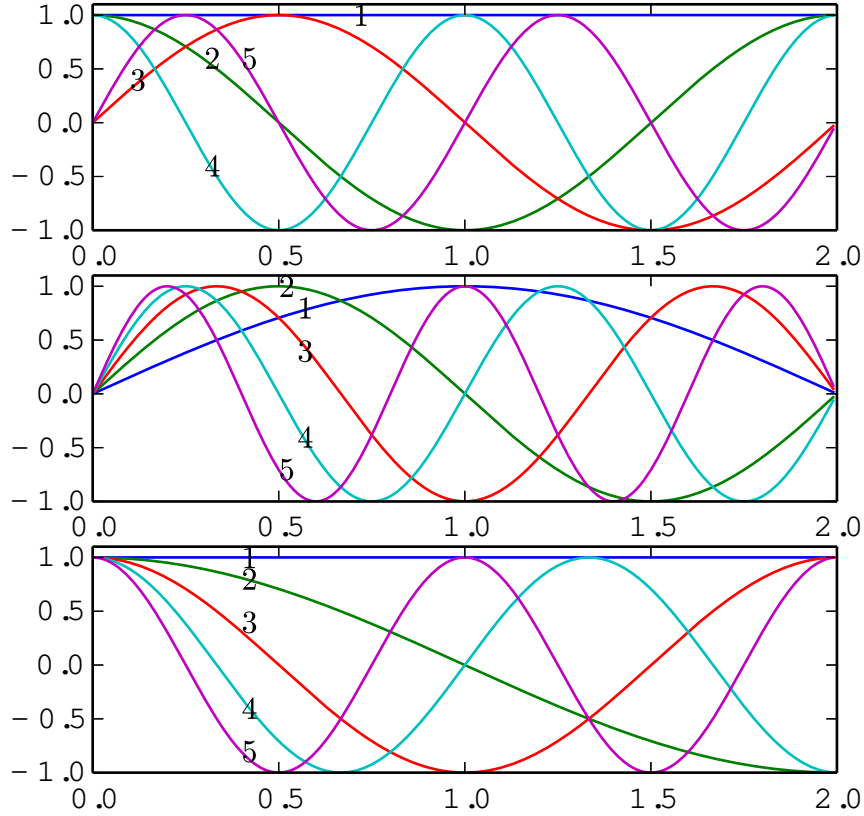


Figure 2. The first five basis functions used in Fourier expansion (upper), sine expansion (middle), and cosine expansion (lower) in the interval $[x_0, x_0 + 2L]$ with $x_0 = 0$ and $L = 1$.

The basis function $b_k(x)$ used in the Fourier expansion have the properties $b_k(x) = b_k(x_0 + 2L)$. Therefore Fourier expansion works best for function that satisfy $h(x_0) = h(x_0 + 2L)$. For a functions that do not satisfies this condition,

i.e., a function with $h(x_0) \neq h(x_0 + 2L)$, the function can still be considered as a periodic function with period $2L$ but having discontinuity points at the interval boundary. It is well known that Gibbs oscillations appear near discontinuity points, which can be inner points in the interval or at the interval boundaries.

The basis functions $b_k(x)$ used in the sine expansion have the properties $b_k(x_0) = b_k(x_0 + 2L) = 0$. Therefore this expansion works best for functions that satisfy $h(x_0) = h(x_0 + 2L) = 0$. For functions that do not satisfy this condition, there will be Gibbs oscillations near the boundaries when approximated by using the sine expansion. Examples are shown in Fig. 3.

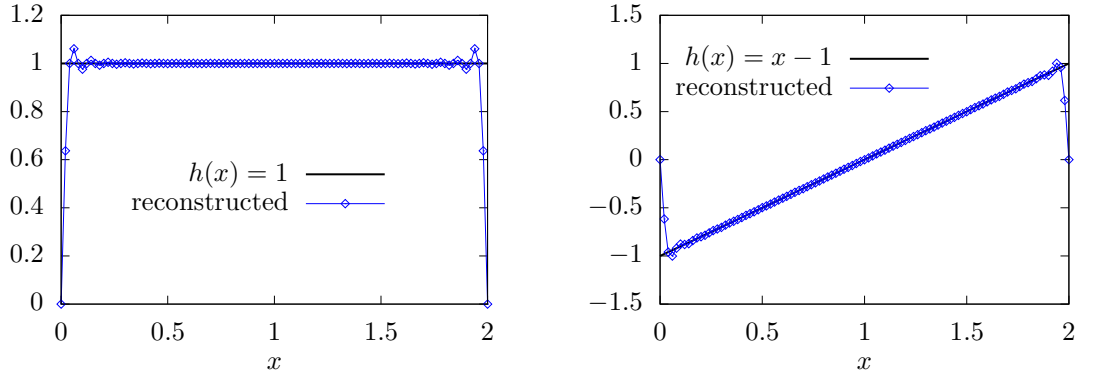


Figure 3. Left: constant function $h(x)=1$ approximated by using the sine expansion. Right: linear function $h(x) = x - 1$ approximated by using the sine expansion. Gibbs oscillation appears near the boundaries, where $h(x)$ does not satisfy the condition $h(x_0) = h(x_0 + L) = 0$ ($x_0 = 0$ and $L = 2$ for this case). The expansion coefficients H_k are obtained via the discrete sine transform (55) with number of sampling point $N = 50$. The reconstruction formula is given by $h(x) = \frac{2}{N} \sum_{k=1}^{N-1} H_k \sin\left(\frac{k\pi x}{L}\right)$.

Similarly, the basis functions $b_k(x)$ used in the cosine expansion have the properties $b'_k(x_0) = b'_k(x_0 + 2L) = 0$. Therefore this expansion works best for functions that satisfy $h'(x_0) = h'(x_0 + 2L) = 0$. For functions that do not satisfy this condition, there will be Gibbs oscillations near the boundaries when approximated by using the cosine expansion (to be verified numerically).

Next, let us discuss the sine and cosine transformation. Figure 4 illustrates the grids used in the following discussion.

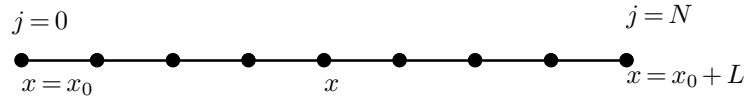


Figure 4. Grid indexes start from 0 and ends at N . $h_0 = h_N = 0$, i.e., $h(x_0) = h(x_0 + L) = 0$.

6.1 Traditional definition of discrete Sine Transform (DST)

There are several slightly different types of Discrete Sine Transforms (DST). One form I saw in W. Press's numerical recipe book is given by

$$H_k = \sum_{j=0}^{N-1} h_j \sin\left(\frac{k\pi j}{N}\right), \quad (55)$$

where $h_0 = h_N = 0$ are assumed. This form can be obtained by replacing DFT's exponential function $\exp(2\pi i k j / N)$ by $\sin(\pi k j / N)$. The inverse sine transformation is given by (I did not derive this, but had numerically verified that this transform recovers the original data if applied after the sine transform (55) (code at /home/yj/project_new/test_space/sine_expansion/t2.f90))

$$h_j = \frac{2}{N} \sum_{k=0}^{N-1} H_k \sin\left(\frac{k\pi j}{N}\right), \quad (56)$$

which is identical with the forward sine transformation except for the normalization factor $2 / N$. (The terms of index 0 in both Eq. (55) and (56) can be dropped since they are zero. They are kept to make the representation look more similar to the DFT, where the index begin from zero.) Replacing j / N in Eq. (56) by $(x - x_0) / L$, we obtain the reconstructing function

$$h(x) = \frac{2}{N} \sum_{k=0}^{N-1} H_k \sin\left(\frac{k\pi(x - x_0)}{L}\right). \quad (57)$$

We need a fast method for computing the above DST. All fast methods finally need to make use of the fast method used in the computation of DFT, i.e., FFT. To make use of FFT, we need to define the DST in a way that the DST can be easily connected to the DFT so that FFT can be easily applied. A standard way of making this easy is to define the DST via the DFT of an odd extension of the original data. Next, let us discuss this.

6.2 Define DST via DFT

Let us introduce the Discrete Sine Transform (DST) by odd extending a given real number sequence and then using the DFT of the extended data to define DST. There are several slightly different ways of odd extending a given sequence and thus different types of DST. Given a $n = 3$ real number sequence (a, b, c) , one frequently adopted odd extension is $(0, a, b, c, 0, -a, -b, -c, 0)$. This odd extension is illustrated in Fig. 5.

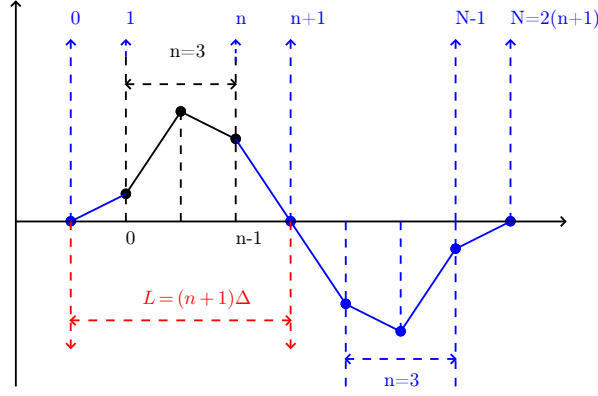


Figure 5. A frequently used way of defining the odd extension of a sequence of real numbers. The black points are original data (with index $0, 1, \dots, n-1$). The blue points are newly introduced, together with the original points, generating an odd periodic sequence of numbers with index $0, 1, \dots, N$, where $N = 2(n+1)$. The points used in the DFT are $0, 1, 2, \dots, N-1$. The period of these data is $2L$ with $L = (n+1)\Delta$, where Δ is the spacing between points.

As illustrated in Fig. 5, after the odd extension, the total number of points is $N+1$ with $N = 2(n+1)$. Then DFT use the N points with index $j = 0, 1, \dots, N-1$ as input. Since the input are real and odd symmetric sequence, the output of this DFT is an odd sequence of purely imaginary numbers. Next, let us prove this. The DFT in this case is given by

$$H_k = \sum_{j=0}^{N-1} h'_j \exp\left(-i \frac{2\pi k j}{N}\right), \quad (58)$$

where h'_j is the odd extension of the original data h_j . For $j = 1, 2, \dots, n$, the relation between h'_j and h_j is given by

$$h'_j = h_{j-1}, \quad (59)$$

For $j = n+2, \dots, N-1$, the relation is given by

$$h'_j = -h_{2n+1-j}. \quad (60)$$

Noting that $h'_0 = 0$ and $h'_{n+1} = 0$, then expression (58) is written as

$$H_k = 0 + \sum_{j=1}^n h'_j \exp\left(-\frac{2\pi i}{N} k j\right) + 0 + \sum_{j=n+2}^{N-1} h'_j \exp\left(-\frac{2\pi i}{N} k j\right). \quad (61)$$

Using $N = 2(n+1)$, the above expression is written as

$$H_k = \sum_{j=1}^n h'_j \exp\left(-\frac{\pi i}{(n+1)} k j\right) + \sum_{j=n+2}^{2n+1} h'_j \exp\left(-\frac{\pi i}{(n+1)} k j\right) \quad (62)$$

Using the relations (59) and (60) to replace h'_j by h_j , the above expression is written

$$H_k = \sum_{j=1}^n h_{j-1} \exp\left(-\frac{\pi i}{(n+1)} k j\right) - \sum_{j=n+2}^{2n+1} h_{2n+1-j} \exp\left(-\frac{\pi i}{(n+1)} k j\right). \quad (63)$$

Change the definition of the dummy index j in the above summation to make it in the conventional range $[0: n-1]$, the above expression is written as

$$H_k = \sum_{j=0}^{n-1} h_j \exp\left(-\frac{\pi i}{(n+1)} k(j+1)\right) - \sum_{j=0}^{n-1} h_{n-1-j} \exp\left(-\frac{\pi i}{(n+1)} k(j+n+2)\right).$$

Defining $j' = n-1-j$ to replace the dummy index in the second summation, the above expression is written as

$$\begin{aligned} H_k &= \sum_{j=0}^{n-1} h_j \exp\left(-\frac{\pi i}{(n+1)} k(j+1)\right) - \sum_{j'=n-1}^0 h_{j'} \exp\left(-\frac{\pi i}{(n+1)} k(2n+1-j')\right) \\ &= \sum_{j=0}^{n-1} h_j \exp\left(-\frac{\pi i}{(n+1)} k(j+1)\right) - \sum_{j=0}^{n-1} h_j \exp\left(-\frac{\pi i}{(n+1)} k(2n+1-j)\right) \\ &= \sum_{j=0}^{n-1} h_j \exp\left(-\frac{\pi i}{(n+1)} k(j+1)\right) - \sum_{j=0}^{n-1} h_j \exp\left(-\frac{\pi i}{(n+1)} k(-j-1)\right) \\ &= -2i \sum_{j=0}^{n-1} h_j \sin\left(\frac{\pi}{(n+1)} k(j+1)\right), \end{aligned} \quad (64)$$

which is a purely imaginary number. Expression (64) also indicates H_k has the following symmetry

$$H_{N-k} = -H_k, \quad (65)$$

i.e. odd symmetry. Therefore only half of the data for H_k with $k = 0, 1, \dots, N-1$ need to be stored, namely H_k with $k = 0, 1, \dots, N/2$. Expression (64) indicates that H_0 and $H_{N/2}$ are definitely zero and thus do not need to be stored. Then the remaining data to be stored are H_k with $k = 1, 2, \dots, N/2-1$, i.e. $k = 1, 2, \dots, n$. Following the convention of making the index of H_k in the range $[0: n-1]$, we define $H'_k = H_{k+1}$. Then

$$H'_k = -2i \sum_{j=0}^{n-1} h_j \sin\left(\frac{\pi}{(n+1)} (k+1)(j+1)\right) \quad (66)$$

with $k = 0, 1, 2, \dots, n-1$, which is the index that we prefer. Finally, the so-called type-I Discrete Sine Transform (DST-I) is defined based on Eq. (66) via

$$Y_k = \frac{H'_k}{-i} = 2 \sum_{j=0}^{n-1} h_j \sin\left(\frac{\pi}{(n+1)}(k+1)(j+1)\right), \quad (67)$$

with $k = 0, 1, 2, \dots, n-1$. This is the RODFT00 transform defined in the FFTW library.

6.3 Reconstruct original function using DST

From the above derivation, we know that the DST, Y_k , is related to the DFT, H_k , by $Y_k = H_{k+1} / (-i)$, and thus the meaning of Y_k is in principle clear. Next, let us use the DST to reconstruct the original function from which the data are sampled. Since DST is only a special case of DFT, reconstructing the function using DST follows the same procedure used in DFT. In DFT, the function is reconstructed via Eq. (42) (changing to the positive exponent convention), i.e.,

$$h(x) = \frac{1}{N} \left[\sum_{k=0}^{N/2} H_k \exp\left(i \frac{k2\pi x}{2L}\right) + \sum_{k=N/2}^{N-1} H_k \exp\left(i \frac{(k-N)2\pi x}{2L}\right) \right]. \quad (68)$$

where $2L$ is the length of the interval in which the samplings h_j with $j = 0, 1, \dots, N-1$ are made. For our present case, i.e., an odd extension of the original n data, we have $N = 2(n+1)$ and $H_0 = 0$ and $H_{N/2} = 0$. Then Eq. (68) is written as

$$h(x) = \frac{1}{N} \left[0 + \sum_{k=1}^n H_k \exp\left(i \frac{k2\pi x}{2L}\right) + 0 + \sum_{k=n+2}^{2n+1} H_k \exp\left(i \frac{(k-N)2\pi x}{2L}\right) \right]. \quad (69)$$

Using the odd symmetry of H_k , i.e., $H_k = -H_{N-k}$, the above expansion is written as

$$h(x) = \frac{1}{N} \left[\sum_{k=1}^n H_k \exp\left(i \frac{k2\pi x}{2L}\right) - \sum_{k=n+2}^{2n+1} H_{N-k} \exp\left(i \frac{(k-N)2\pi x}{2L}\right) \right]. \quad (70)$$

Define $k' = 2n+2-k$, and note that $N = 2(n+1)$, then the above expression is written as

$$h(x) = \frac{1}{2(n+1)} \left[\sum_{k=1}^n H_k \exp\left(i \frac{k2\pi x}{2L}\right) - \sum_{k'=n}^1 H_{k'} \exp\left(i \frac{(-k')2\pi x}{2L}\right) \right]. \quad (71)$$

i.e.,

$$h(x) = \frac{1}{2(n+1)} \sum_{k=1}^n H_k \left[\exp\left(i \frac{k2\pi x}{2L}\right) - \exp\left(-i \frac{k2\pi x}{2L}\right) \right], \quad (72)$$

which is simplified as

$$h(x) = \frac{1}{2(n+1)} \sum_{k=1}^n 2i H_k \sin\left(\frac{k2\pi x}{2L}\right). \quad (73)$$

As a convention, we prefer that the summation index begins from 0 and ends at $n-1$. Then expression (73) is written as

$$h(x) = \frac{1}{2(n+1)} \sum_{k=0}^{n-1} 2i H_{k+1} \sin\left(\frac{(k+1)2\pi x}{2L}\right)$$

Using the relation between DST and DFT, i.e., $H_{k+1} = -iY_k$, the above equation is written as

$$h(x) = \frac{1}{n+1} \sum_{k=0}^{n-1} Y_k \sin\left(\frac{(k+1)2\pi x}{2L}\right). \quad (74)$$

This is the formula for constructing the continuous function using the DST data. This formula is an expansion over the basis functions $\sin[(k+1)\pi x / L]$ with the DST Y_k acting as the expansion coefficients. Therefore the direct meaning of the DST, Y_k , is that they are the expansion coefficients when using $\sin(k\pi x / L)$ as the basis functions to approximate a function in the domain $[0, L]$. From Fig. 5, we know that the interval length L is given by $L = (n+1)\Delta$, where Δ is the uniform spacing between the original n sampling points.

6.3.1 Inverse DST

Evaluate the function in Eq. (74) at $x_j = (j+1)\Delta$ with $j = 0, 1, \dots, n-1$, then we obtain

$$\begin{aligned} h(x_j) &= \frac{1}{n+1} \sum_{k=0}^{n-1} Y_k \sin\left(\frac{(k+1)2\pi(j+1)\Delta}{2L}\right) \\ &= \frac{1}{n+1} \sum_{k=0}^{n-1} Y_k \sin\left(\frac{(k+1)\pi(j+1)}{(n+1)}\right). \end{aligned} \quad (75)$$

It can be proved that $h(x_j)$ in the above equation exactly recover h_j used in defining the DST Y_k . Therefore Eq. (75) is the Inverse Discrete Sine Transform.

6.4 Discrete Cosine transform

to be written

7 Misc content

7.1 Nonlinear process

The trigonometric identity

$$2\cos^2(\omega t) = 1 + \cos(2\omega t), \quad (76)$$

indicates that nonlinear interaction between two waves of the same frequency generates zero frequency component and its second harmonic. The process of generating of second-harmonic is also called frequency doubling.

7.2 Aliasing errors

Signals that are not band-limited usually contains all frequencies and thus do not satisfy the condition required by the sampling theorem (i.e., $H(f) = 0$ for $|f| > 1/(2\Delta)$). In this case, for any given N data, we can still calculate its DFT by using Eq. (34). However the results obtained are meaningful only when H_n approaches zero as the frequency approaches $-1/(2\Delta)$ from above and approaches $1/(2\Delta)$ from below, i.e., only when the results obtained are consistent with the assumption used to obtain the results (the assumption is that $H(f) = 0$ for $|f| > 1/(2\Delta)$). When the results obtained do not satisfy the above condition, then it indicates that the “aliasing errors” have contributed to the results. We can reduce the aliasing errors by increasing the sampling frequency. The aliasing errors can be reduced but can not be completely removed for a non-band-limited signal.

In signal processing and related disciplines, **aliasing** is an effect that causes different signals to become indistinguishable (or *aliases* of one another) when sampled. It also often refers to the distortion or artifact that results when a signal reconstructed from samples is different from the original continuous signal.

An alias is a false lower frequency component that appears in sampled data acquired at too low a sampling rate.

Aliasing errors are hard to detect and almost impossible to remove using software. The solution is to use a high enough sampling rate, or if this is not possible, to use an anti-aliasing filter in front of the analog-to-digital converter (ADC) to eliminate the high frequency components before they get into the data acquisition system.

When a digital image is viewed, a reconstruction is performed by a display or printer device, and by the eyes and the brain. If the image data is processed in some way during sampling or reconstruction, the reconstructed image will differ from the original image, and an alias is seen.

Aliasing can be caused either by the sampling stage or the reconstruction stage; these may be distinguished by calling sampling aliasing (*prealiasing*) and reconstruction aliasing (*postaliasing*).

In video or cinematography, temporal aliasing results from the limited frame rate, and causes the [wagon-wheel effect](#), whereby a spoked wheel appears to rotate too slowly or even backwards. Aliasing has changed its apparent frequency of rotation. A reversal of direction can be described as a [negative frequency](#). Temporal aliasing frequencies in video and cinematography are determined by the frame rate of the camera, but the relative intensity of the aliased frequencies is determined by the shutter timing (exposure time) or the use of a temporal aliasing reduction filter during filming.

7.3 Relation between Fourier series coefficients and DFTs

In the above, we go through the process “Fourier series→Fourier transformation →DFT”, which corresponds to going from the discrete case (Fourier series) to the continuous case (Fourier transformation), and then back to the discrete case (DFT). Since both Fourier series and DFT are discrete in frequency, it is instructive to examine the relation between the Fourier coefficient c_n and the DFT H_n . The Fourier coefficient of $h(t)$ is given by Eq. (93), i.e.,

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} h(t) e^{in2\pi t/T} dt, \quad (77)$$

which can be equivalently written

$$c(f_n) = \frac{1}{T} \int_{-T/2}^{T/2} h(t) e^{i2\pi t f_n} dt, \quad (78)$$

where $f_n = n/T$. On the other hand, if $h(t)$ is sampled with sampling rate $f_s = 1/\Delta$, then the number of sampling points per period is $N = T/\Delta$. Then the frequency at which the Fourier transform $H(f)$ is evaluated in getting the DFT [Eq. (36)] is written

$$f_n = \frac{n}{\Delta N} = \frac{n}{T}, \quad (79)$$

which is identical to the frequency to which the Fourier coefficient c_n corresponds. Using $f_n = n/(\Delta N)$ in Eq. (78), we obtain

$$\begin{aligned} c(f_n) &= \frac{1}{T} \int_{-T/2}^{T/2} h(t) \exp\left(\frac{i2\pi t n}{\Delta N}\right) dt. \\ &\approx \frac{1}{T} \Delta \sum_{j=0}^{N-1} h_j \exp\left(\frac{2\pi i}{N} n j\right) \end{aligned} \quad (80)$$

$$\begin{aligned} &= \frac{1}{T} \Delta H_n \\ &= \frac{1}{N} H_n \end{aligned} \quad (81)$$

i.e., dividing the DFT H_n by N gives the corresponding Fourier expansion coefficient c_n . We can further use Eqs. (29) and (30) to recover the Fourier coefficients in terms of trigonometric functions cosine and sine,

Note that the approximation in Eq. (80) becomes an exact relation if the largest frequency contained in $h(t)$ is less than $1/(2\Delta)$.

A Efficient method of computing DFT: Fast Fourier Transformation (FFT) algorithm (not finished)

The DFT is defined by Eq. (34), i.e.,

$$H_j \equiv \sum_{k=0}^{N-1} W^{kj} h_k, \quad (82)$$

where $W = \exp(2\pi i / N)$. Equations (82) indicates that the DFT is the multiplication of a transformation matrix $M_{kj} \equiv W^{kj}$ with a column vector h_k , where the transformation matrix M_{kj} is symmetric and called DFT matrix. In the matrix form, the DFT is written as

$$\begin{pmatrix} H_1 \\ H_2 \\ H_3 \\ H_4 \\ \vdots \\ H_{N-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & W^1 & W^2 & W^3 & \dots & W^{N-1} \\ 1 & W^2 & W^4 & W^6 & \dots & W^{2(N-1)} \\ 1 & W^3 & W^6 & W^9 & \dots & W^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W^{N-1} & W^{2(N-1)} & W^{3(N-1)} & \dots & W^{(N-1)(N-1)} \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ \vdots \\ h_{N-1} \end{pmatrix} \quad (83)$$

If directly using the definition in Eq. (83) to compute DFT, then a matrix multiplication need to be performed, which requires $O(N^2)$ operations. Here the powers of W are assumed to be pre-computed, and we define ‘‘an operation’’ as a multiplication followed by an addition.

The Fast Fourier Transformation (FFT) algorithm manage to reduce the complexity of computing the DFT from $O(N^2)$ to $O(N \log_2 N)$ by factoring the DFT matrix M_{kj} into a product of sparse matrices.

The original paper on FFT is Cooley and Tukey’s paper: An algorithm for the machine calculation of complex Fourier series, which turns out to be a very concise paper.

Suppose N is a composite, i.e., $N = r_1 \cdot r_2$. Then the indices in Eq. (82) can be expressed as

$$j = j_1 r_1 + j_0, \quad j_0 = 0, 1, \dots, r_1 - 1, \quad j_1 = 0, 1, \dots, r_2 - 1 \quad (84)$$

$$k = k_1 r_2 + k_0, \quad k_0 = 0, 1, \dots, r_2 - 1, \quad k_1 = 0, 1, \dots, r_1 - 1 \quad (85)$$

Then, one can write

$$H(j_0, j_1) = \sum_{k_0} \sum_{k_1} h(k_0, k_1) W^{(j k_1 r_2 + j k_0)}, \quad (86)$$

Noting that

$$W^{j k_1 r_2} = W^{j_1 r_1 k_1 r_2 + j_0 k_1 r_2} = W^{j_0 k_1 r_2} \quad (87)$$

Eq. (86) is written as

$$H(j_0, j_1) = \sum_{k_0} \left(\sum_{k_1} h(k_0, k_1) W^{(j_0 k_1 r_2)} \right) W^{(j k_0)}, \quad (88)$$

where the inner sum over k_1 is independent of j_1 and can be defined as a new array

$$h_1(j_0, k_0) = \sum_{k_1} h(k_0, k_1) W^{(j_0 k_1 r_2)}. \quad (89)$$

Then

$$H(j_0, j_1) = \sum_{k_0} h_1(j_0, k_0) W^{(j_1 r_1 + j_0) k_0}, \quad (90)$$

There are N elements in the array h_1 , each requiring r_1 operations, giving a total of $N r_1$ operation to obtain h_1 . Similarly, it takes $N r_2$ operations to calculate H from h_1 . Therefore, this two-step algorithm requires a total of $N(r_1 + r_2)$ operations.

B Fourier series→Fourier transformation

B.1 From discrete spectrum to continuous spectrum

The Fourier series discussed above indicates that a periodic function is composed of discrete spectrum and is written as

$$h(t) = \sum_{n=-\infty}^{\infty} c_n \exp\left(i \frac{n 2\pi}{T} t\right), \quad (91)$$

where T is the period of $h(t)$. The n th term of the above Fourier series corresponds to a harmonic of frequency

$$f_n = \frac{n}{T}, \quad (92)$$

and the expansion coefficient c_n is given by

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} h(t) \exp\left(-i \frac{n2\pi}{T} t\right) dt. \quad (93)$$

In terms of f_n , the coefficient in Eq. (93) is written

$$c_n = c(f_n) = \frac{1}{T} \int_{-T/2}^{T/2} h(t) \exp(-i2\pi f_n t) dt \quad (94)$$

In terms of f_n , the Fourier series in Eq. (91) is written

$$\begin{aligned} h(t) &= \sum_{n=-\infty}^{\infty} c(f_n) e^{i2\pi t f_n} \\ &= T \sum_{n=-\infty}^{\infty} c(f_n) e^{i2\pi t f_n} \frac{1}{T} \end{aligned} \quad (95)$$

Note that $c(f_n) e^{i2\pi t f_n}$ is the value of function $c(f) e^{i2\pi t f}$ at $f = f_n$. Further note that the interval between f_n and f_{n+1} is $1/T$. Thus the above summation is the rectangular formula for numerically calculating the integration $\int_{-\infty}^{\infty} c(f) e^{i2\pi t f} df$. Therefore, equation (95) can be approximately written as

$$h(t) \approx T \int_{-\infty}^{\infty} c(f) e^{i2\pi t f} df, \quad (96)$$

which will become exact when the interval $1/T \rightarrow 0$, i.e., $T \rightarrow \infty$. Therefore, for the case $T \rightarrow \infty$, the Fourier series exactly becomes

$$h(t) = T \int_{-\infty}^{\infty} c(f) e^{i2\pi t f} df, \quad (97)$$

where $c(f)$ is given by Eq. (94), i.e.,

$$c(f) = \frac{1}{T} \int_{-T/2}^{T/2} h(t) e^{-i2\pi t f} dt. \quad (98)$$

Note that the function $h(t)$ given in Eq. (97) is proportional to $Tc(f)$ while the function $c(f)$ given in Eq. (98) is proportional to $1/T$. Since $T \rightarrow \infty$, it is desired to eliminate the T and $1/T$ factors in Eqs. (97) and (98), which can be easily achieved by defining a new function

$$H(f) = Tc(f). \quad (99)$$

Then equations (97) and (98) are written as

$$h(t) = \int_{-\infty}^{\infty} H(f) e^{i2\pi t f} df, \quad (100)$$

$$H(f) = \int_{-T/2}^{T/2} h(t) e^{-i2\pi t f} dt = \int_{-\infty}^{\infty} h(t) e^{-i2\pi t f} dt, \quad (101)$$

Equations (100) and (101) are the Fourier transformation pairs discussed in the next section.

B.2 Fourier transformation

As discussed above, the Fourier transformation of a function $h(t)$ is given by

$$H(f) = \int_{-\infty}^{\infty} h(t) e^{i2\pi ft} dt. \quad (102)$$

Once the Fourier transformation $H(f)$ is known, the original function $h(t)$ can be reconstructed via

$$h(t) = \int_{-\infty}^{\infty} H(f) e^{-i2\pi ft} df. \quad (103)$$

[Note that the signs in the exponential of Eq. (100) and (101) are opposite. Which one should be minus or positive is actually a matter of convention because a trivial variable substitution $f' = -f$ can change the sign between minus and positive. Proof. In terms of f' , Eq. (103) is written

$$\begin{aligned} h(t) &= \int_{\infty}^{-\infty} H(-f') e^{i2\pi t f'} d(-f'), \\ &= \int_{-\infty}^{\infty} H(-f') e^{i2\pi t f'} df', \end{aligned} \quad (104)$$

Define

$$\begin{aligned} \bar{H}(f') &\equiv H(-f') \\ &= \int_{-\infty}^{\infty} h(t) e^{-i2\pi t f'} dt. \end{aligned} \quad (105)$$

Then Eq. (104) is written

$$h(t) = \int_{-\infty}^{\infty} \bar{H}(f') e^{i2\pi t f'} df' \quad (106)$$

The signs in the exponential of Eqs. (105) and (106) are opposite to Eqs. (102) and (103), respectively.]

[Some physicists prefer to use the angular frequency $\omega \equiv 2\pi f$ rather than the frequency f to represent the Fourier transformation. Using ω , equations. (102) and (103) are written, respectively, as

$$H(\omega) = \int_{-\infty}^{\infty} h(t) e^{i\omega t} dt, \quad (107)$$

$$h(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(\omega) e^{-i\omega t} d\omega, \quad (108)$$

where we see that the asymmetry between the Fourier transformation and its inverse is more severe in this representation: besides the opposite-sign in the exponents, there is also a $1/2\pi$ factor difference between the Fourier transformation and its inverse. Whether the $1/2\pi$ factor appears at the forward transformation or inverse one is actually a matter of convention. The only requirement is that the product of the two factors in the forward and inverse transformation is equal to $1/2\pi$. To obtain a more symmetric pair, one can

adopt a factor $1/\sqrt{2\pi}$ at both the forward and inverse transformation. The representation in Eqs. (102) and (103) is adopted in this note. But we should know how to change to the ω representation when needed.]

$$\begin{aligned}
 & \left[\int_{-\infty}^{\infty} \frac{dh(t)}{dt} e^{-i2\pi t f} dt = h(t) e^{-i2\pi t f} \Big|_{-\infty}^{+\infty} - \int_{-\infty}^{\infty} h(t) \frac{d e^{-i2\pi t f}}{dt} dt = \right. \\
 & \left. - \int_{-\infty}^{\infty} h(t) \frac{d e^{-i2\pi t f}}{dt} dt = i2\pi f \int_{-\infty}^{\infty} h(t) e^{-i2\pi t f} dt = i2\pi f H(f) \right] \quad (109)
 \end{aligned}$$

B.3 Numerical computation of Fourier transformation

Next, consider how to numerically compute the Fourier transformation of a function $h(t)$. A simple way is to use the rectangle formula to approximate the integration in Eq. (102), i.e.,

$$H(f) \approx \Delta \sum_{j=-\infty}^{\infty} h_j \exp(i2\pi f j \Delta), \quad (110)$$

where $h_j = h(t_j)$ and $t_j = j\Delta$ with $j = \dots, -2, -1, 0, 1, 2, \dots$. Note Eq. (110) is an approximation, which will become exact if $\Delta \rightarrow 0$. In practice, we can sample $h(t)$ only with a nonzero Δ . Therefore Eq. (110) is usually an approximation. Do we have some rules to choose a suitable Δ so that Eq. (110) can become a good approximation or even an exact relation? This important question is answered by the sampling theorem, which states that a suitable Δ to make Eq. (110) exact is given by $\Delta \leq 1/(2f_c)$, where f_c is the largest frequency contained in $h(t)$ (i.e., $H(f) = 0$ for $|f| > f_c$).

B.4 Sampling theorem

In computational and experimental work, we know only a list of values $h(t_j)$ sampled at discrete values of t_j . Let us suppose that $h(t)$ is sampled with uniform interval between consecutive points:

$$h_j = h(t_j), t_j = j\Delta, j = \dots, -2, -1, 0, 1, 2, \dots \quad (111)$$

The sampling rate is defined by $f_s = 1/\Delta$. The sampling theorem states that: If the Fourier transformation of function $h(t)$, $H(f)$, has the following property

$$H(f) = 0 \quad \text{for } |f| \geq f_c \quad (112)$$

then sampling $h(t)$ with the sampling rate $f_s \geq 2f_c$ (i.e., $\Delta \leq 1/(2f_c)$) will completely determine $h(t)$, which is given explicitly by the formula

$$h(t) = \Delta \sum_{j=-\infty}^{\infty} h_j \frac{\sin[2\pi f_c(t - j\Delta)]}{\pi(t - j\Delta)}. \quad (113)$$

We will not concern us here with the proof of the sampling theorem and simply start working with Eq. (113) to derive the concrete expression for the Fourier transformation of $h(t)$. Substituting the expression (113) for $h(t)$ into the Fourier transformation (102), we obtain the explicit form of the Fourier transformation of $h(t)$:

$$\begin{aligned}
H(f) &\equiv \int_{-\infty}^{\infty} h(t) \exp(i2\pi ft) dt, \\
&= \Delta \sum_{j=-\infty}^{\infty} h_j \int_{-\infty}^{\infty} \frac{\sin[2\pi f_c(t-j\Delta)]}{\pi(t-j\Delta)} \exp(i2\pi ft) dt, \\
&= \Delta \sum_{j=-\infty}^{\infty} h_j \exp(i2\pi f j \Delta) \int_{-\infty}^{\infty} \frac{\sin[2\pi f_c(t-j\Delta)]}{\pi(t-j\Delta)} \exp[i2\pi f(t-j\Delta)] dt, \\
&= \Delta \sum_{j=-\infty}^{\infty} h_j \exp(i2\pi f j \Delta) \int_{-\infty}^{\infty} \frac{\sin[2\pi f_c \tau]}{\pi \tau} \exp[i2\pi f \tau] d\tau. \tag{114}
\end{aligned}$$

With the help of Wolfram Mathematica, the integration in Eq. (114) is evaluated analytically, giving

$$\int_{-\infty}^{\infty} \frac{\sin[2\pi f_c \tau]}{\pi \tau} \exp(i2\pi f \tau) d\tau = \begin{cases} 0 & \text{For } |f| > f_c \\ 1 & \text{For } |f| < f_c \end{cases}. \tag{115}$$

Using this, Eq. (114) is written

$$H(f) = \begin{cases} 0 & \text{for } |f| > f_c \\ \Delta \sum_{j=-\infty}^{\infty} h_j \exp(i2\pi f j \Delta) & \text{for } |f| \leq f_c \end{cases}, \tag{116}$$

which shows that $H(f) = 0$ for $|f| > f_c$, which is consistent with the assumption of sampling theorem, i.e., $H(f)$ has the property given in Eq. (112). The second line of Eq. (116) is identical to Eq. (110) except that Eq. (116) in this case is exact while Eq. (110) is only approximate. In other words, if $\Delta \leq 1/(2f_c)$, then the Fourier transformation is exactly given by Eq. (116), where f_c is the largest frequency contained in $h(t)$ (i.e., $H(f) = 0$ for $|f| > f_c$).

C 2D Fourier series

Compared with Eq. (1) that uses the trigonometric functions, Fourier series (15) and (16), which is expressed in terms of the complex exponential function $e^{in\pi x/L}$, is more compact. The convenience introduced by the complex exponential function is more obvious when we deal with multiple-dimensional cases. For example, a two-dimensional function $G(x, y)$ can be expanded as Fourier series about x ,

$$G(x, y) = \sum_{m=-\infty}^{\infty} c_m(y) e^{im\pi x/L_x}, \tag{117}$$

where $2L_x$ is the period of G in x direction. The expansion coefficients $c_m(y)$ can be further expanded as Fourier series about y ,

$$c_m(y) = \sum_{n=-\infty}^{\infty} (c_{mn} e^{in\pi y/L_y}), \quad (118)$$

where $2L_y$ is the period of G in y direction, and the coefficients c_{mn} is given by

$$\begin{aligned} c_{mn} &= \frac{1}{2L_y} \int_{-L_y}^{L_y} \left[\frac{1}{2L_x} \int_{-L_x}^{L_x} G(x, y) e^{-im\pi x/L_x} dx \right] e^{-in\pi y/L_y} dy. \\ &= \frac{1}{4L_x L_y} \int_{-L_y}^{L_y} \int_{-L_x}^{L_x} G(x, y) e^{-im\pi x/L_x - in\pi y/L_y} dx dy. \end{aligned} \quad (119)$$

Using Eq. (118) in Eq. (117), we obtain

$$G(x, y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} c_{mn} e^{in\pi y/L_y + im\pi x/L_x}, \quad (120)$$

Equations (120) and (119) give the two-dimensional Fourier series of $G(x, y)$.

C.1 For 2D real-valued functions

The formula for expanding a real-valued two-dimensional function $G(x, y)$ in terms of basis functions $\cos\left(\frac{m\pi x}{L_x} + \frac{n\pi y}{L_y}\right)$ and $\sin\left(\frac{m\pi x}{L_x} + \frac{n\pi y}{L_y}\right)$ can be readily recovered from Eqs. (120) and (119). For notation ease, define

$$\alpha = \frac{m\pi x}{L_x} + \frac{n\pi y}{L_y}. \quad (121)$$

Then Eq. (120) is written as

$$\begin{aligned} G(x, y) &= \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} c_{mn} [\cos\alpha + i \sin\alpha] \\ &= \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \left(\frac{1}{4L_x L_y} \int_{-L_y}^{L_y} \int_{-L_x}^{L_x} G(x, y) \cos\alpha dx dy \right) [\cos\alpha + i \sin\alpha] \\ &\quad + \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \left(\frac{1}{4L_x L_y} \int_{-L_y}^{L_y} \int_{-L_x}^{L_x} G(x, y) \sin\alpha dx dy \right) [-i \cos\alpha + \sin\alpha], \end{aligned}$$

Sine $G(x, y)$ is assumed to be real-valued, the imaginary parts of the above expression will cancel each other. Therefore, the above expansion is simplified to

$$G(x, y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \left[\left(\frac{1}{4L_x L_y} \int_{-L_y}^{L_y} \int_{-L_x}^{L_x} G(x, y) \cos \alpha dx dy \right) \cos \alpha + \left(\frac{1}{4L_x L_y} \int_{-L_y}^{L_y} \int_{-L_x}^{L_x} G(x, y) \sin \alpha dx dy \right) \sin \alpha \right]. \quad (122)$$

This is a compact Fouier expansion for 2D real-valued function. Furthermore, noting that (m, n) term is equal to $(-m, -n)$ term, the above expansion can be further reduced to

$$G(x, y) = \sum_{n=-\infty}^{\infty} \sum_{m=0}^{\infty} G_{nm}^c \cos \alpha + G_{nm}^s \sin \alpha, \quad (123)$$

with

$$G_{00}^c = \frac{1}{4L_x L_y} \int_{-L_y}^{L_y} \int_{-L_x}^{L_x} G(x, y) dx dy. \quad (124)$$

and the other coefficients given by

$$G_{mn}^c = \frac{1}{2L_x L_y} \int_{-L_y}^{L_y} \int_{-L_x}^{L_x} G(x, y) \cos \alpha dx dy. \quad (125)$$

$$G_{mn}^s = \frac{1}{2L_x L_y} \int_{-L_y}^{L_y} \int_{-L_x}^{L_x} G(x, y) \sin \alpha dx dy. \quad (126)$$

Here the range of m is reduced to $[0; +\infty]$. In this case, we have an edge case, G_{00}^c , that needs special treatment. We see that allowing the index runing from $-\infty$ to $-\infty$ has the advantage of that there are no edge cases that needs special treatment.

C.2 2D real-valued Fourier series derived directly from real-valued trigonometric functions expansion– to be deleted, because there is an easier way to do this, as is given by the above section

We see that the extension of Fourier series from one-dimension case to two-dimension case is straightforward when expressed in terms of the complex exponential function $e^{in\pi x/L}$. However, if we use $\sin(m\pi x/L_x)$, $\cos(m\pi x/L_x)$, $\sin(n\pi y/L_y)$, and $\cos(n\pi y/L_y)$ as basis functions, the derivation of the two-dimensional Fourier series of $G(x, y)$ is a little complicated (product-to-sum trigonometric identities are involved to simplify the results). Let's see the

derivation. A two-dimensional function $G(x, y)$ can be first expanded as Fourier series about x ,

$$G(x, y) = \sum_{m=1}^{\infty} a_m(y) \cos\left(\frac{m\pi}{L_x}x\right) + \sum_{m=1}^{\infty} b_m(y) \sin\left(\frac{m\pi}{L_x}x\right), \quad (127)$$

(the zero-frequency component is dropped, will take it back later) and then the two coefficients $a_m(y)$ and $b_m(y)$ can be further expanded as Fourier series about y ,

$$a_m(y) = \sum_{n=1}^{\infty} a_{mn}^{(a)} \cos\left(\frac{n\pi}{L_y}y\right) + \sum_{n=1}^{\infty} b_{mn}^{(a)} \sin\left(\frac{n\pi}{L_y}y\right), \quad (128)$$

$$b_m(y) = \sum_{n=1}^{\infty} a_{mn}^{(b)} \cos\left(\frac{n\pi}{L_y}y\right) + \sum_{n=1}^{\infty} b_{mn}^{(b)} \sin\left(\frac{n\pi}{L_y}y\right), \quad (129)$$

(the zero-frequency component is dropped, will take it back later) Substituting Eq. (128) and (129) into Eq. (127), we obtain

$$\begin{aligned} G(x, y) = & \sum_{m=1}^{\infty} \left[\sum_{n=1}^{\infty} a_{mn}^{(a)} \cos\left(\frac{n\pi}{L_y}y\right) + \sum_{n=1}^{\infty} b_{mn}^{(a)} \sin\left(\frac{n\pi}{L_y}y\right) \right] \cos\left(\frac{m\pi}{L_x}x\right) \\ & + \sum_{m=1}^{\infty} \left[\sum_{n=1}^{\infty} a_{mn}^{(b)} \cos\left(\frac{n\pi}{L_y}y\right) + \sum_{n=1}^{\infty} b_{mn}^{(b)} \sin\left(\frac{n\pi}{L_y}y\right) \right] \sin\left(\frac{m\pi}{L_x}x\right). \end{aligned} \quad (130)$$

Using the product-to-sum trigonometric identities, equation (130) is written

$$\begin{aligned} G(x, y) = & \frac{1}{2} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} a_{mn}^{(a)} \left[\cos\left(\frac{m\pi}{L_x}x + \frac{n\pi}{L_y}y\right) + \cos\left(\frac{m\pi}{L_x}x - \frac{n\pi}{L_y}y\right) \right] \\ & + \frac{1}{2} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} b_{mn}^{(a)} \left[\sin\left(\frac{m\pi}{L_x}x + \frac{n\pi}{L_y}y\right) + \sin\left(\frac{m\pi}{L_x}x - \frac{n\pi}{L_y}y\right) \right] \\ & + \frac{1}{2} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} a_{mn}^{(b)} \left[\sin\left(\frac{m\pi}{L_x}x + \frac{n\pi}{L_y}y\right) - \sin\left(\frac{m\pi}{L_x}x - \frac{n\pi}{L_y}y\right) \right] \\ & + \frac{1}{2} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} b_{mn}^{(b)} \left[\cos\left(\frac{m\pi}{L_x}x - \frac{n\pi}{L_y}y\right) - \cos\left(\frac{m\pi}{L_x}x + \frac{n\pi}{L_y}y\right) \right] \end{aligned}$$

which can be organized as

$$\begin{aligned} G(x, y) = & \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{1}{2} [a_{mn}^{(a)} - b_{mn}^{(b)}] \cos\left(\frac{m\pi}{L_x}x + \frac{n\pi}{L_y}y\right) \\ & + \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{1}{2} [b_{mn}^{(a)} + a_{mn}^{(b)}] \sin\left(\frac{m\pi}{L_x}x + \frac{n\pi}{L_y}y\right) \end{aligned}$$

$$\begin{aligned}
& + \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{1}{2} [a_{mn}^{(a)} + b_{mn}^{(b)}] \cos\left(\frac{m\pi}{L_x}x - \frac{n\pi}{L_y}y\right) \\
& + \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{1}{2} [b_{mn}^{(a)} - a_{mn}^{(b)}] \sin\left(\frac{m\pi}{L_x}x - \frac{n\pi}{L_y}y\right)
\end{aligned} \tag{131}$$

The coefficients appearing above are written as

$$a_m(y) = \frac{1}{L_x} \int_{-L_x}^{-L_x} G(x, y) \cos\left(\frac{m\pi}{L_x}x\right) dx \tag{132}$$

$$b_m(y) = \frac{1}{L_x} \int_{-L_x}^{-L_x} G(x, y) \sin\left(\frac{m\pi}{L_x}x\right) dx \tag{133}$$

$$a_{mn}^{(a)} = \frac{1}{L_x} \frac{1}{L_y} \int_{-L_y}^{-L_y} \int_{-L_x}^{-L_x} G(x, y) \cos\left(\frac{m\pi}{L_x}x\right) \cos\left(\frac{n\pi}{L_y}y\right) dx dy \tag{134}$$

$$a_{mn}^{(b)} = \frac{1}{L_x} \frac{1}{L_y} \int_{-L_y}^{-L_y} \int_{-L_x}^{-L_x} G(x, y) \cos\left(\frac{m\pi}{L_x}x\right) \sin\left(\frac{n\pi}{L_y}y\right) dx dy \tag{135}$$

$$b_{mn}^{(a)} = \frac{1}{L_x} \frac{1}{L_y} \int_{-L_y}^{-L_y} \int_{-L_x}^{-L_x} G(x, y) \sin\left(\frac{m\pi}{L_x}x\right) \cos\left(\frac{n\pi}{L_y}y\right) dx dy \tag{136}$$

$$b_{mn}^{(b)} = \frac{1}{L_x} \frac{1}{L_y} \int_{-L_y}^{-L_y} \int_{-L_x}^{-L_x} G(x, y) \sin\left(\frac{m\pi}{L_x}x\right) \sin\left(\frac{n\pi}{L_y}y\right) dx dy \tag{137}$$

Noting that $b_{m,-n}^{(b)} = -b_{m,n}^{(b)}$, and $a_{m,-n}^{(b)} = -a_{m,n}^{(b)}$ and $\sin(0) = 0$, then Eq. (131) can be written as

$$\begin{aligned}
G(x, y) &= \sum_{m=1}^{\infty} \sum_{n=-\infty}^{\infty} \frac{1}{2} [a_{mn}^{(a)} - b_{mn}^{(b)}] \cos\left(\frac{m\pi}{L_x}x + \frac{n\pi}{L_y}y\right) \\
&+ \sum_{m=1}^{\infty} \sum_{n=-\infty}^{\infty} \frac{1}{2} [b_{mn}^{(a)} + a_{mn}^{(b)}] \sin\left(\frac{m\pi}{L_x}x + \frac{n\pi}{L_y}y\right).
\end{aligned} \tag{138}$$

The coefficients can be further written as

$$\begin{aligned}
c_{mn} &\equiv \frac{1}{2} [a_{mn}^{(a)} - b_{mn}^{(b)}] \\
&= \frac{1}{2} \frac{1}{L_x} \frac{1}{L_y} \int_{-L_y}^{-L_y} \int_{-L_x}^{-L_x} G(x, y) \left[\cos\left(\frac{m\pi}{L_x}x\right) \cos\left(\frac{n\pi}{L_y}y\right) - \right. \\
&\quad \left. \sin\left(\frac{m\pi}{L_x}x\right) \sin\left(\frac{n\pi}{L_y}y\right) \right] dx dy \\
&= \frac{1}{2} \frac{1}{L_x} \frac{1}{L_y} \int_{-L_y}^{-L_y} \int_{-L_x}^{-L_x} G(x, y) \cos\left(\frac{m\pi}{L_x}x + \frac{n\pi}{L_y}y\right) dx dy.
\end{aligned} \tag{139}$$

$$\begin{aligned}
 s_{mn} &\equiv \frac{1}{2}[b_{mn}^{(a)} + a_{mn}^{(b)}] \\
 &= \frac{1}{2} \frac{1}{L_x} \frac{1}{L_y} \int_{-L_y}^{-L_y} \int_{-L_x}^{-L_x} G(x, y) \left[\sin\left(\frac{m\pi}{L_x}x\right) \cos\left(\frac{n\pi}{L_y}y\right) + \right. \\
 &\quad \left. \cos\left(\frac{m\pi}{L_x}x\right) \sin\left(\frac{n\pi}{L_y}y\right) \right] dx dy \\
 &= \frac{1}{2} \frac{1}{L_x} \frac{1}{L_y} \int_{-L_y}^{-L_y} \int_{-L_x}^{-L_x} G(x, y) \sin\left(\frac{m\pi}{L_x}x + \frac{n\pi}{L_y}y\right) dx dy. \tag{140}
 \end{aligned}$$

Then comes the drudgery to handle the special cases of $m = 0$ and/or $n = 0$. The final results are identical to Eqs. (123)-(126). This kind of derivation is also discussed in my notes on mega code.

D Details on FFT codes provided by the Numerical recipes book[2]

The following is about a specific FFT subroutine provided by the Numerical recipes book[2]. This is not a general case.

The input and output of the DFT are usually complex numbers. In the implementation of FFT algorithm provided by Numerical recipes book[2], to avoid using complex numbers, the algorithm adopts the real number representation of the complex numbers. In this scheme, two elements of a real number array are used to store one complex number. To store a complex array “cdata” of length N , we will need a real number array “rdata” of length $2N$. The first elements of array “rdata” will contain the real part of “cdata(1)”, the second elements of “rdata” will contain the imaginary part of “cdata(1)”, and so on.

To test the correctness of the above statement, I generated a real number array with length $N = 2 \times 2^8$ by using a random generating routine and calculate the DFT of the array with two methods. The real array generated by the random generator are considered to be a real number representation of a complex array with length $N / 2$. Using the real array as the input of the FFT routine (the code in ~/project_new/fft). To check the correctness of my understanding of the input and output of the FFT, I manually convert the real number of length N to a complex array with length $N / 2$, and use directly the summation in Eq. (?) to calculate the DFT. The output I got is obviously a complex array with length $N / 2$. Then I manually convert the complex array to a real number array of length N and plot the output in Fig. 6 with dashed line. The results in Fig. 6 indicates the results given by the FFT and the naive method used by me agree with each other well. This proves that my understanding of the input and output of FFT (especially the storage arrangement) is correct.

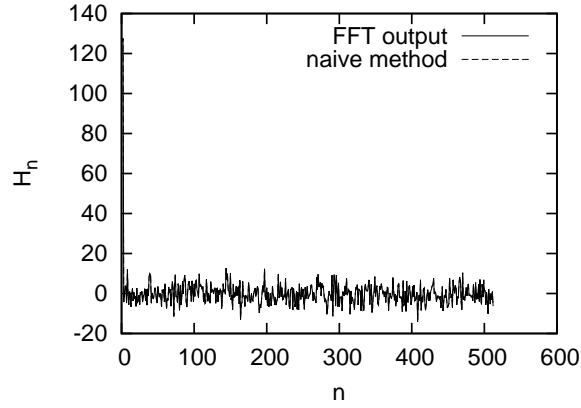


Figure 6. The output of a FFT routine (solid line) agrees well with those (dashed line) calculated by directly evaluate the summation in Eq. (?) (the latter is coded by me). The agreement indicates my understanding of the output of FFT (especially the storage arrangement) is correct. Here the time domain data is generated by a random generating routine.

To clearly show the output of FFT, we recover the real and imaginary part of DFT from the output of FFT and plots the data as a function of their corresponding frequency. The results are given in Fig. 7.

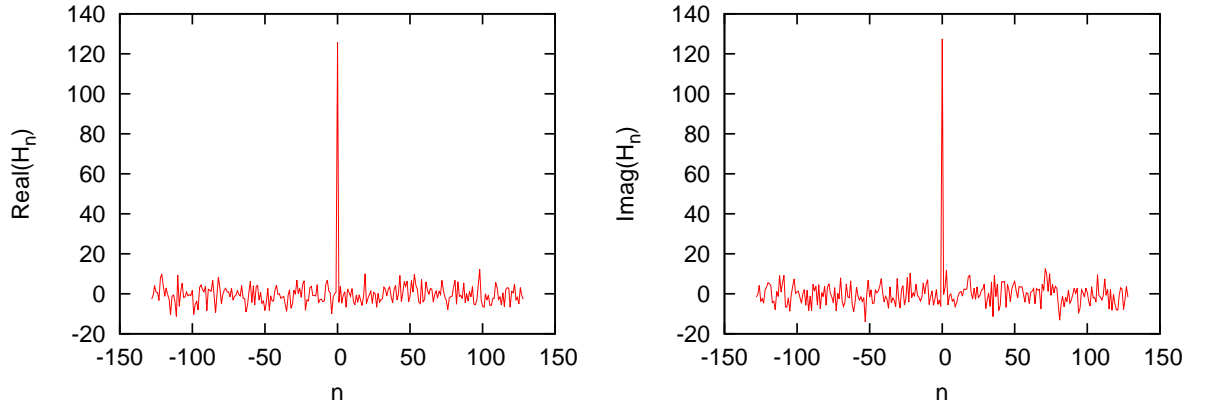


Figure 7. The real (left) and imaginary (right) part of the discrete Fourier transformation as a function of the frequency. The point $n = -128$ corresponds to frequency $-f_s/2$ while the point $n = 128$ corresponds to frequency $+f_s/2$, where $f_s = 1/\Delta$ with Δ is the sample interval in time domain. The time domain data is the same as used in Fig. 6. Why is there a spike at zero frequency?

D.1 Computing Fourier integrals using FFT (not finished, to be deleted)

Consider the calculation of the following integral:

$$I(\omega) = \int_a^b e^{i\omega t} h(t) dt. \quad (141)$$

Divide the interval $[a, b]$ into M uniform sub-intervals and define

$$\Delta = \frac{b-a}{M}, t_j = a + \Delta j, h_j = h(t_j), j = 0, 1, 2, \dots, M \quad (142)$$

Then the integration in Eq. (141) can be approximated as

$$I(\omega) \approx \Delta \sum_{j=0}^{M-1} h_j \exp(i\omega t_j). \quad (143)$$

Define $\omega_m = 2\pi m / (M\Delta)$ with integer m and $-M/2 < m < M/2$. Consider the calculation of $I(\omega_m)$. Using Eq. (143), we obtain

$$\begin{aligned} I(\omega_m) &= \Delta \sum_{j=0}^{M-1} h_j \exp[i\omega_m(a + \Delta j)] \\ &= \Delta e^{i\omega_m a} \sum_{j=0}^{M-1} h_j \exp(i\omega_m \Delta j) \\ &= \Delta e^{i\omega_m a} \sum_{j=0}^{M-1} h_j \exp\left(i \frac{2\pi m}{M} j\right) \\ &= \Delta e^{i\omega_m a} H_m \end{aligned} \quad (144)$$

$$= \Delta e^{i\omega_m a} [\text{DFT}(h_0, h_1, h_2, \dots, h_{M-1})]_m. \quad (145)$$

Equation (145) indicates the value of the integration $I(\omega_m)$ can be obtained by calculating the discrete Fourier transformation of h_j . However, as discussed in Ref. [2], equation (145) is not recommended for practical use because the oscillatory nature of the integral will make Eq. (145) become systematically inaccurate as ω increases. Next, consider a new method, in which $h(t)$ is expanded as

$$h(t) \approx \sum_{j=0}^M h_j \psi\left(\frac{t-t_j}{\Delta}\right) + \sum_{j=\text{endpoints}} h_j \varphi_j\left(\frac{t-t_j}{\Delta}\right) \quad (146)$$

Apply the integral operator $\int_a^b dt \exp(i\omega t)$ to both sides of Eq. (146), we obtain

$$\begin{aligned} \int_a^b h(t) e^{i\omega t} dt &\approx \sum_{j=0}^M h_j \int_a^b \psi\left(\frac{t-t_j}{\Delta}\right) e^{i\omega t} dt + \\ &\sum_{j=\text{endpoints}} h_j \int_a^b \varphi_j\left(\frac{t-t_j}{\Delta}\right) e^{i\omega t} dt. \end{aligned} \quad (147)$$

Make the change of variables $s = (t - t_j) / \Delta$ in the first integral and $s = (t - a) / \Delta$ in the second integral, the above equation is written as

$$\int_a^b h(t) e^{i\omega t} dt \approx \Delta \sum_{j=0}^M h_j \int_a^b \psi(s) e^{i\omega(\Delta s + t_j)} ds + \Delta \sum_{j=\text{endpoints}} h_j \int_a^b \varphi_j(s - j) e^{i\omega(\Delta s + a)} ds \quad (148)$$

Define $\theta = \omega\Delta$ and make use of $t_j = a + j\Delta$, the above equation is written as

$$\int_a^b h(t) e^{i\omega t} dt \approx \Delta e^{i\omega a} \sum_{j=0}^M h_j e^{i\theta j} \int_a^b \psi(s) e^{i\theta s} ds + \Delta e^{i\omega a} \sum_{j=\text{endpoints}} h_j \int_a^b \varphi_j(s - j) e^{i\theta s} ds \quad (149)$$

Define

$$W(\theta) = \int_a^b \psi(s) e^{i\theta s} ds \quad (150)$$

$$\alpha_j(\theta) = \int_a^b \varphi_j(s - j) e^{i\theta s} ds \quad (151)$$

Then Eq. (149) is written as

$$\int_a^b h(t) e^{i\omega t} dt \approx \Delta e^{i\omega a} \left[W(\theta) \sum_{j=0}^M h_j e^{i\theta j} + \sum_{j=\text{endpoints}} h_j \alpha_j(\theta) \right]. \quad (152)$$

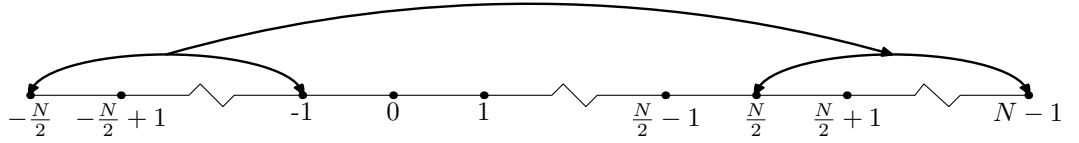


Figure 8. Older version of Fig. ?, created by Metapost, the new version is created by the vector graphic editor in TeXmacs.

Bibliography

- [1] *A Guided Tour of Mathematical Physics*. Samizdat Press, 1994.
- [2] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in Fortran 77*. Cambridge University Press, Cambridge, UK, 1992.