B.Tech Mini-Project Report

Understanding Fractional Fourier Transform

Under the guidance of

(Dr. Bosanta R Boruah)

Submitted by:
Aman Kumar Saini
(Roll no.180121003)

Department of Physics
Indian Institute of Technology, Guwahati
April 2021

Contents

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Dr. Bosanta R Boruah, for providing his invaluable guidance, comments, and suggestions throughout the project and constantly motivate me to work harder.

Introduction

This chapter provides a brief overview of classical Fourier transform and fractional Fourier transform. Essential properties of fractional Fourier transform have also been listed.

1.1 Ordinary Fourier transform

Fourier transform becomes a particular case of more generalized fractional Fourier transform. Fourier transform is a mathematical tool used in functional analyses and has many applications. Mathematically, Fourier transform is an integral transform that decomposes a function defined in space or time domain into a function defined in the spatial or temporal frequency domain. Some Fourier transform properties have been mentioned below, which ultimately motivate the definition of fractional Fourier transform(FrFT).

Definition*: On a suitable functional space, e.g., $L^2(R)$, Fourier transform, and its inverse operating on a function f are defined as*

$$F(\epsilon) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} f(x)e^{-i\epsilon x}dx, \qquad f(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} F(\epsilon)e^{i\epsilon x}\,d\epsilon \tag{1}$$

Generally, f is a time-dependent function, and ϵ denotes frequency, but it is not limited to time-dependent functions. $F(\epsilon)$ is the representation of the function f in the frequency domain. (see fig. 1)

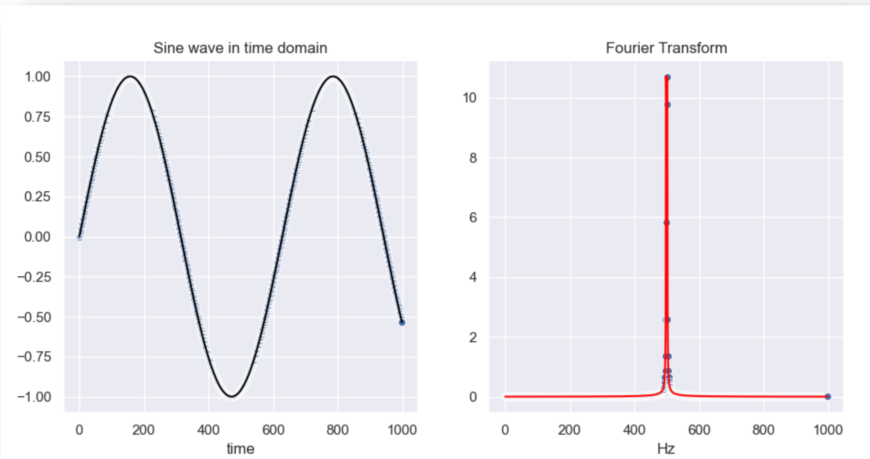

Figure1: *Implementation of Fourier transform on a sine wave in the time domain, resulting in a Dirac delta function in the frequency domain.*

If F^2 represents twice implementation of Fourier transform on a function such that $F^2 = FF$, similarly $F^3 = FFF$, and so on. It is readily verified that $(F^2 f)(x) = f(-x)$, $(F^3 f)(\epsilon) = F(-\epsilon)$, and $(F^4 f)(x) = f(x)$. Apparently, for a ϵ Z, we can say that F^a is a rotation in the (x, ϵ) plane of angle $\alpha = a\pi/2$. With this interpretation, FrFT defines the rotation when *a* is **not** an integer.

1.2 Fractional Fourier transform

Fractional Fourier transform has many interpretations pertaining to different applications. Continuing with our previous definition, the a^{th} order FrFT is the a^{th} power of the Fourier operator. It extends to the values of *a* which are not integers. Consequently, FrFT facilitates rotation in-between time and frequency domain. Hence, the frequency domain is one particular case from a set of continuous fractional Fourier domains. FrFT is a one-parameter family of transforms. Linear canonical transform, which has three parameters, is further a generalization of fractional Fourier transform.

Definition*: The a^{th} order fractional Fourier transform of a function f(x) is defined as a linear integral transform with kernel* $K_a(\epsilon, x)$*:*

$$(F_a f)(\epsilon) \coloneqq f_a(\epsilon) = \int_{-\infty}^{\infty} K_a(\epsilon, x) f(x) dx \tag{2}$$

Where the kernel

$$K_a(\epsilon, x) = C_a \exp[i\pi(\cot\alpha\, \epsilon^2 - 2\csc\alpha\, \epsilon x + \cot\alpha\, x^2)] \tag{3}$$

And $\alpha = a\pi/2\,, \; C_a = \sqrt{1 - i\cot\alpha}$

Equation (3) is periodic in *a (or* α *)* with period 4 (or 2π). hus, we can limit our discussion to the interval *a* ϵ *[-2, 2)*. In general, *a* is a real number, and *f* is a finite value function. Many basic properties of FrFT become apparent from this definition, like:

1. *F^0 = I (Identity Operator)*
2. *F^1 = F (Ordinary Fourier transform)*
3. *F^2 = P (Parity Operator)*
4. *F^3 = P.F*
5. *F^4 = F^0 = I*

6. $F^{4j+a} = F^{4j'+a}$ *(General form of (5) , where j and j' are arbitrary integers)*

However, the case that brings out the fundamental importance of FrFT is when *a* is not an integer.

For small a(hence small α) such that $|a| > 0$, the kernel can be written as

$$K_a(\epsilon, x) = \frac{e^{-i\pi \mathrm{sgn}(\alpha)/4}}{\sqrt{|\alpha|}} \exp[i\pi(\epsilon - x)^2/\alpha] \tag{4}$$

Having the well-known limit

$$\delta(\epsilon) = \lim_{y \to 0} e^{-\pi/4} \sqrt{\frac{1}{y}}\, e^{i\pi\epsilon^2/y} \tag{5}$$

Therefore, the kernel approaches $\delta(\epsilon - x)$ in the given limit. This behaviour of kernel ensures the continuity of the transform in the limiting cases when *a →0 or a →2n.*

One way to look at plots of FrFT is to see how a function evolves from time-domain (*a = 0*) to frequency domain (*a = 1*), which is a transition from its original form to its Fourier transform (see fig. 2). As the order of FrFT, *a = 0* corresponds to the Identity Operator. *a = 1* corresponds to the Ordinary Fourier transform, which results in a Dirac delta function when the input function is a sine wave. $a \,\epsilon (0,1)$ corresponds to the representation in domains intermediate to time and frequency.

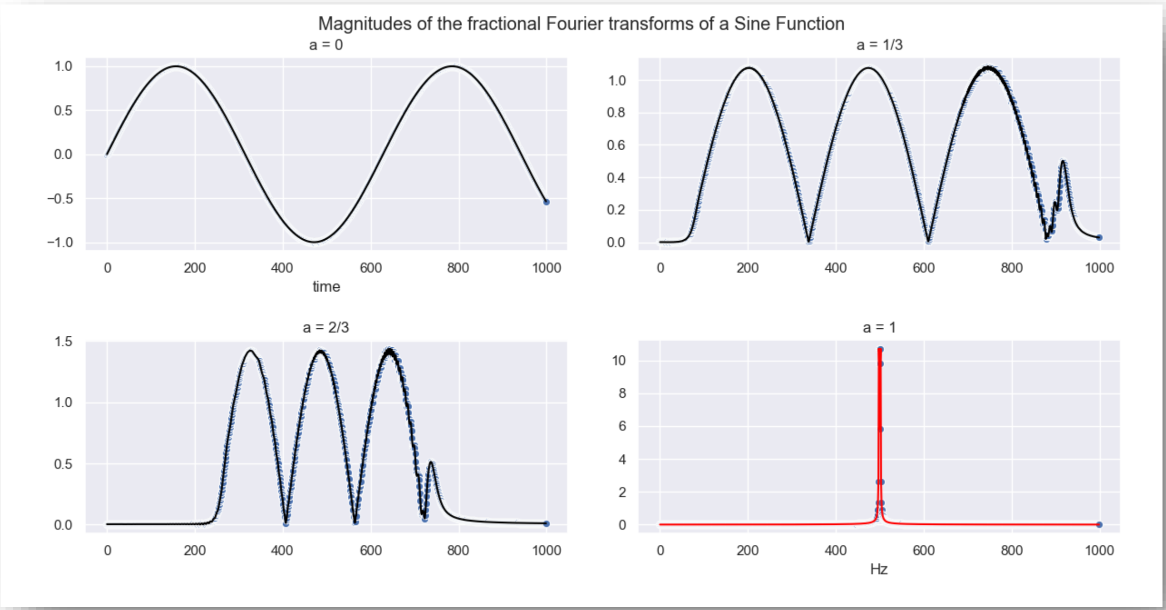

***Figure 2**: Evolution of a sine wave from its original form to the Fourier transform through fractional domains*

FrFT has all the properties of Fourier transform, or one can say they become a particular case of FrFT properties. Listed below are some of the crucial properties of FrFT.

Linearity: FrFT is a linear operator

$$F^a[\sum_k c_k \psi_k(x)] = \sum_k c_k [F^a \psi_k(x)] \tag{6}$$

Index additivity: It is an essential property as it helps to infer F^a as the a^{th} power of the Fourier operator.

$$F^{a_1} F^{a_2} = F^{a_1 + a_2} = F^{a_2} F^{a_1} \tag{7}$$

The popular definition of FrFT is a conclusion of this property.

Commutativity: $F^{a_1} F^{a_2} = F^{a_2} F^{a_1}$

Associativity: $F^{a_1}(F^{a_2} F^{a_3}) = (F^{a_1} F^{a_2}) F^{a_3}$

Unitary: $(F^a)^{-}1 = (F^a)^H = F^{-a}$

Eigenfunctions and Eigenvalues: Classical Fourier trasnform and FrFT share the same eigenfunctions but the eigenvalues of FrFT are the a^{th} power of the eigenvalues of Fourier transform. The eigenfunctions of Fourier transform are famous Hermite-Gaussian functions $\psi_n(u)$ also known as the eigen solutions of the harmonic oscillator in quantum mechanics. Using the eigenvalues and eigenfunctions of FrFT, the spectral decomposition of the kernel of FrFT can be obtained as in [1].

$$K_a(\epsilon, x) = \sum_{n=0}^{\infty} e^{-ian\pi/2} \psi_n(\epsilon) \psi_n(x) \tag{8}$$

Spectral decomposition of the kernel provides a way to calculate FrFT on discrete functions, as discussed later in this letter.

Computation of Fractional Fourier Transform

Digital computation of fractional Fourier transform will be the main focus of this chapter.

2.1 Different Methods to Compute Fractional Fourier Transform

Solving the integral in equation (3) is a big endeavour if one has decided to use all the mathematical tools available out there, but looking at the applications of FrFT, it is clear that a different approach has been employed and numerical integral is at the core of it. Like [2] states, "quadratic exponential, which appears in diffraction theory, requires a large number of numerical integration samples". The problem becomes apparent in the limiting case when *a* is small and close to zero.

One way of evaluating (2) is the use of kernel's spectral decomposition (equation 8). FrFT of a given function *f(u)* from the knowledge of the eigenfunctions and eigenvalues only is obtained by first expanding the function as a linear superposition of the eigenfunctions of the FrFT (which are known to constitute a complete set):

$$f(u) = \sum_{n=0}^{\infty} C_n \psi_n(u) \tag{11}$$

Where, $C_n = \int \psi_n(u) f(u) du$

Applying F^a on both sides of equation (11), we get

$$F^a f(u) = \sum_{n=0}^{\infty} e^{-ian\pi/2} C_n \psi_n(u) \tag{12}$$

Therefore, one can obtain the FrFT of *f(u)* by multiplying the expansion coefficients c_n with $e^{-ian\pi/2}$, and summing the components.

Although such an intuitive approach looks straight forward and may give precise results but the computational expense associated is not good enough for practical purposes.

2.2 Efficient Estimation of Fractional Fourier Transform

Many algorithms present for FrFT computation are generally termed as fast fractional Fourier transform because they are decomposed into more straightforward steps that use a very famous fast Fourier transform(FFT) algorithm for evaluation. The whole principle works on the idea that FrFT is a sub-family of a more broad linear canonical transformations [3]. Canonical transformations can be decomposed into subsequent more straightforward steps, like scaling and ordinary Fourier transform. This fact has been exploited in these algorithms in different ways.

The algorithm that has been used in the report for applications and analysis is discussed in this section.

The motivation of the algorithm is found in rewriting the definition (equations 2,3) as

$$f_a(\epsilon) = C_a e^{-i\pi \tan(\alpha/2)\epsilon^2} \int_{-\infty}^{\infty} e^{i\pi \csc\alpha(\epsilon-x)^2}\left[e^{-i\pi \tan(\alpha/2)x^2} f(x)\right]dx \tag{13}$$

It has been assumed that the function *f* is nonzero only in a finite interval $[-\Delta/2, \Delta/2]$ in all time-frequency directions. This makes the Wigner distribution [4] of *f* becomes zero outside a circle of radius Δ/2. Now, chirp multiplication and chirp convolution of *f* under this consideration makes the Wigner distribution compact and inside a radius Δ, making the FrFT integral zero outside $[-\Delta, \Delta]$. Thus, if the input function *f* has $N = \Delta^2$ samples, then 2Δ samples of the chirp convolution are obtained.

Therefore, The integral is computed at $\epsilon_k = k/2\Delta$ as

$$f_a(\epsilon_k) = \frac{C_a}{2\Delta} e^{i\pi - \epsilon_k{}^2 \tan(\alpha/2)} \sum_{l=-N}^{N-1} g(k-l)h(l) \tag{14}$$

With $x_k = k/2$ and

$$g(k) = e^{i\pi x_k{}^2 \csc\alpha}, \qquad h(k) = e^{i\pi - x_k{}^2 \tan(\alpha/2)} f(x_k)$$

Two operations involved here, multiplication and convolution of chirp functions, involve FFT in their computation and hence give good numerical complexity.

2.3 Numerical Implementation

Direct implementation of the algorithm in [2] does not give expected results. Discussion of the steps that implement this algorithm possible by writing a code in a computer will follow.

1) Required length of the input function

Note that in equation (14), the summation's symmetry is running from *-N* to *N-1* requires the length of the input function to be odd. In the analogy between FFT and fast FrFT, FFT transforms a vector of length N with samples running from *0* to *N-1* and fast FrFT transforms a vector of length 2N with samples running from *-N* to *N-1*. So, the

FFT implementation in the case of FrFT when the order *a* = *±1* will not result as expected in the algorithm [2]. The result will match only when there is a shift over half of the input function length. If *N* is the length of the signal, then the shift has been achieved as [5]

```
shft = rem((0:N-1)+fix(N/2),N)+1;
```

Since the input function length is odd, this operation may introduce some asymmetry.

2) Required interval for the order a

As mentioned earlier, algorithm [2] efficiency decreases substantially due to the kernel's rapid oscillations when *a* is close to 0 or ±2. To work around this problem, we confined the interval for the order such that $a \in [0.5, 1.5]$. In such cases, the integral can be computed directly. If *a* lies outside this interval, we can use the property of *index additivity* $F^{a} = F^{1} F^{a-1}$.

3) Heart of the Algorithm

The algorithm mainly consists of three significant steps:

- Multiplication of input function *f* with a chirp function
- Convolution with a chirp
- Multiply with a chirp

As explained in the appendix of [2], we need to double the number of samples in the input function f to avoid aliasing. Replacement of f with a larger function is done using interpolation. A couple of interpolations can be employed here, but Shannon's interpolation is famous in these applications. The sinc interpolation of f is given by

$$y(x) = \sum_{k=-N}^{N-1} f(x_k)\mathrm{sinc}\big(2\Delta(x - x_k)\big) \qquad (15)$$

It can be evaluated using the input function's convolution with a sinc function obtained using FFT.

Thus if E_c represents a chirp function e^{icx^2}, then the decomposed integral in equation (14) can be written as

$$F^{a}\, y = \frac{C_a}{2\Delta}\, E_c\, F^{-1}\{F[E_d\, y].\, F[E_c y]\} \tag{16}$$

Where $c = \cot\alpha - \csc\alpha = -\tan(\alpha/2)$ and $d = \csc\alpha$. Here *y* is the interpolated sample of the input function *f*.

4) Subsampling of the result

The output function needs to be of the same length as the input function *f* . For this reason, subsampling of the result obtained in step (3) has to be done. Extension of the original function and padding it with zero is a precautionary step to avoid contamination by boundary effects in the middle in the convoluted function in equation (14).

Results and Applications

3.1 Examples

This section will evaluate the fractional Fourier transform using the fast algorithm discussed in chapter 2 for few basic functions like delta function (fig. 3), Gaussian function (fig. 4), and sine waves (fig. 5).

It is interesting to see the delta function's continuous evolution to a constant function as we vary *a* from 0 to 1.

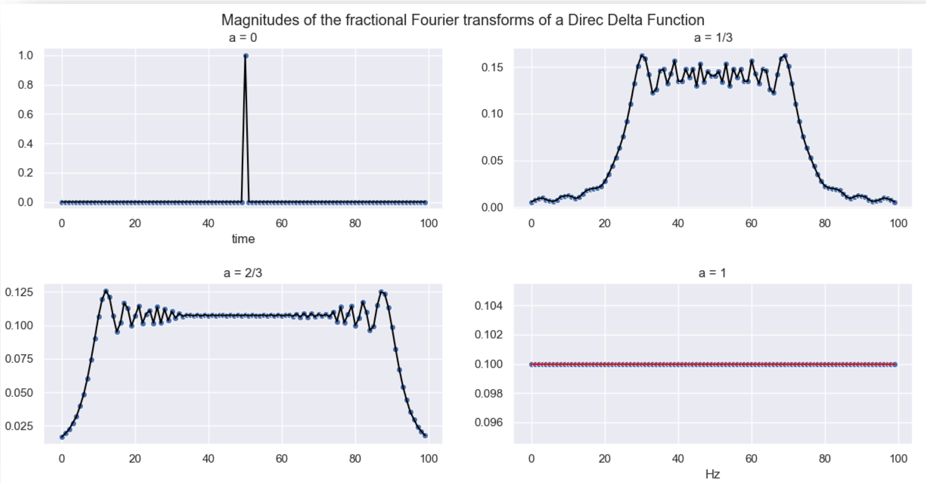

***Figure 3**: Fractional Fourier transforms of a delta function*

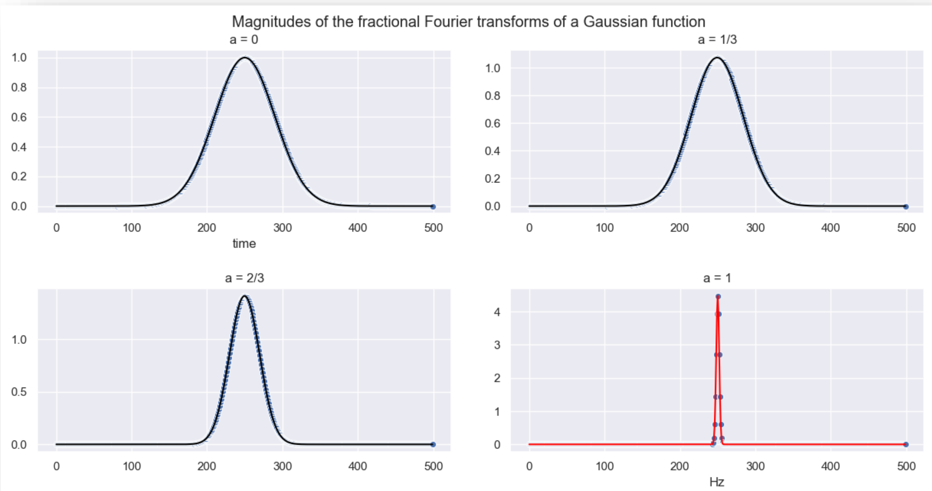

***Figure 4**: Fractional Fourier transforms of a Gaussian function. The Fourier transform(a=1) of a Gaussian function is again a Gaussian with inverse variance, as shown in the figure.*

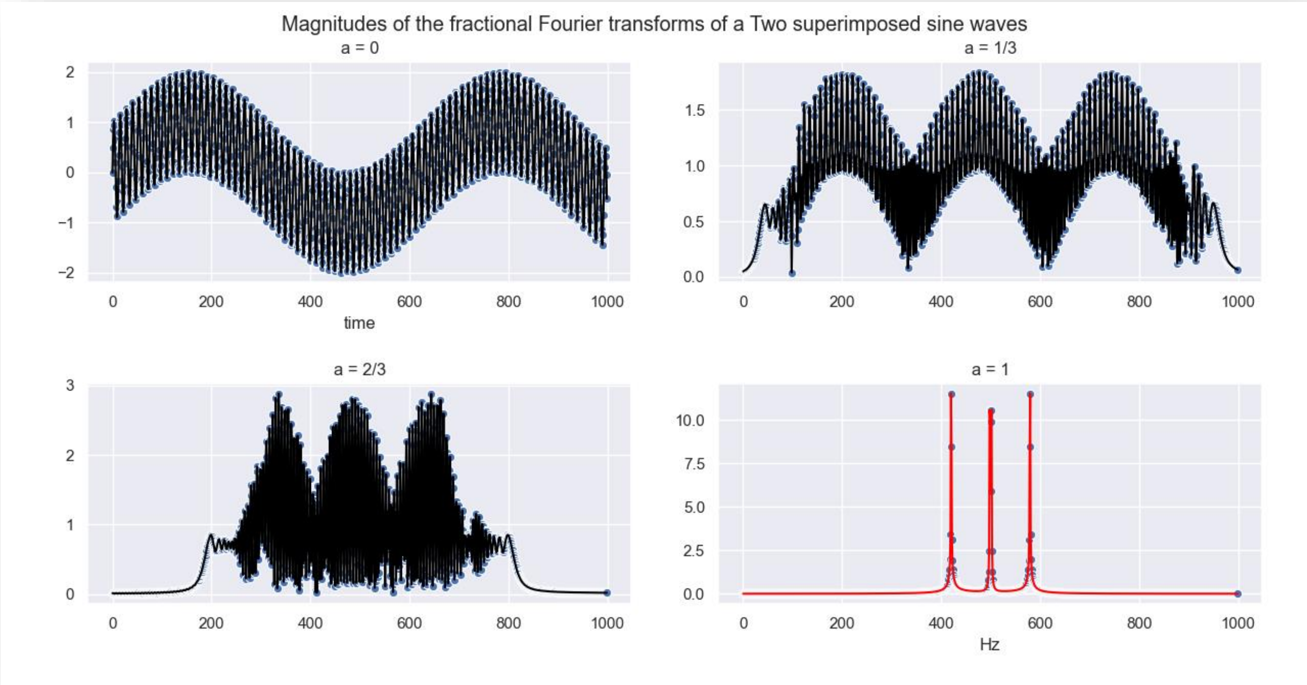

Figure 5: Fractional Fourier transforms of two superimposed sine waves. Involvement of two sine waves with different frequencies(500, 500±100). If there had been only one sine wave, then it would look like something figure 2

3.2 Convolution theorem for fractional Fourier transform

This section develops the convolution theory for fractional Fourier transform.

Unlike the convolution theorem for the Fourier transform, which states that the Fourier transform of a convolution of two functions is the pointwise product of their Fourier transforms, the same for FrFT does not result as friendly or practical. As explained in [7], every integral transformation T can be theoretically be associated with a convolution operation $\star$, such that

$$\mathrm{T}(\mathrm{f}\star\mathrm{g}) = \mathrm{T}(\mathrm{f})\mathrm{T}(\mathrm{g}) \tag{17}$$

Before introducing a new convolution for FrFT, let us first show that the traditional convolution does not give satisfactory results.

Let f and g be two functions defined in the space of integrable functions, and their convolution h be denoted as $f * g$ such that

$$\mathrm{h}(\mathrm{x}) = (\mathrm{f} * \mathrm{g})(\mathrm{x}) = \int_{-\infty}^{\infty} f(t)g(x-t)dt \tag{18}$$

Then the fractional Fourier transform of h [10], denoted by H_α

$$H_\alpha = |\sec\alpha| e^{-i(u^2/2)\tan\alpha} \int_{-\infty}^{\infty} F_\alpha(v) g[(u-v)\sec\alpha] \; x \; e^{-i(v^2/2)cot\alpha} dv \qquad (19)$$

The new convolution theorem for fractional Fourier transform will now be discussed. It preserves the property of traditional convolution and is easier to implement.

Notation:

$d(\alpha) = (\cot\alpha)/2$, $\qquad c(\alpha) = \sqrt{1 - i\cot\alpha}$ $\qquad$ *and* $\qquad \tilde{f}(x) = f(x)e^{idx^2}$

$b(\alpha) = \sec\alpha$

Definition*: For any two function f and g, the convolution is defined as*

$$h(x) = (f\star g)(x) = \frac{c}{\sqrt{2\pi}} e^{-idx^2} \left(\tilde{f} * \tilde{g}\right)(x) \qquad (20)$$

*where * is the traditional convolution used for Fourier transform.*

Theorem: *Let H_α , G_α , F_α denote the FrFT of h, g, f , respectively and h(x)=(f ★g)(x). Then*

$$H_\alpha(u) = F_\alpha(u) G_\alpha(u) e^{-idu^2} \qquad (21)$$

Proof: *Using equation (2) of FrFT definition, we get*

$$H_\alpha(u) = \frac{c}{\sqrt{2\pi}} \int_{-\infty}^{\infty} h(t) \; e^{i[d(t^2+u^2)-2dbut]} dt$$

$$= \frac{c}{2\pi} \int_{-\infty}^{\infty} e^{i[d(t^2+u^2)-2dbut]} e^{-idt^2} dt \; x \int_{-\infty}^{\infty} f(x) \, e^{idx^2} g(t-x) e^{id(t-x)^2} dx$$

$$= \frac{c^2}{2\pi} \iint_{-\infty}^{\infty} f(x) g(t-x) \exp\{i[d(t^2+u^2) - 2dbut + 2dx^2 - 2dtx]\} \, dxdt$$

Let $t - x = v$, we get

$$H_\alpha(u) = \frac{c^2}{2\pi} \iint_{-\infty}^{\infty} f(x) g(v) \exp[id(x^2+u^2+v^2) - 2idbu(x+v)] \, dxdv$$

$$= \frac{c^2\, e^{-idu^2}}{2\pi} \int_{-\infty}^{\infty} \mathrm{f(x)e^{id(x^2+u^2)-2idbux}\, dx \; x} \int_{-\infty}^{\infty} \mathrm{g(v)e^{id(v^2+u^2)-2idbuv}\, dv}$$

$$= \mathrm{e^{-idu^2} F_\alpha(u) G_\alpha(u)}$$

Hence proved for (21)

Experimental results

Here we have applied the new convolution theorem for fractional Fourier transform on some basic functions.

Example 1. In this example, we have computed the FrFT of a sine wave and a rectangular pulse. Later verified the convolution theorem graphically. In the figure, graphs (f) and (g) represent the expressions in (21)

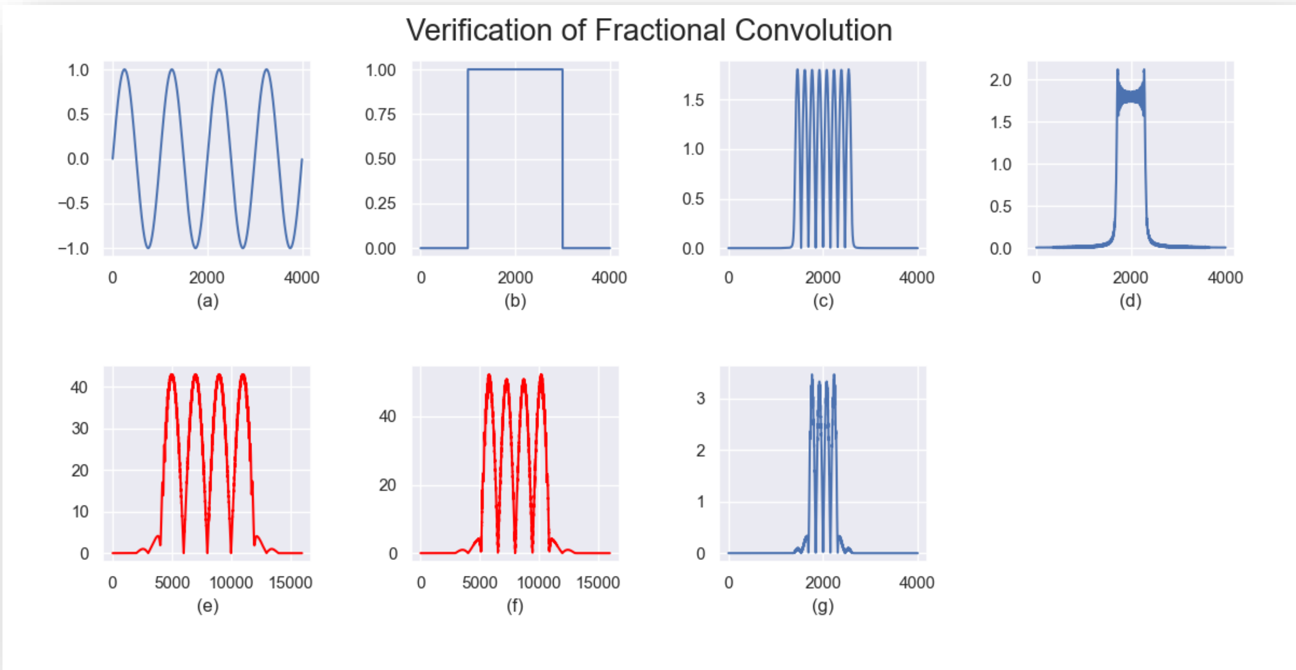

Figure 6: *(a) A sine wave f . (b) A rectangular pulse g , (a) and (b) are input functions for the verification. (c) FrFT of f for a=0.8. (d) FrFT of g for a=0.8. (e) Fractional Convolution h of f and g. (f) FrFT of h at a=0.8. (g) Pointwise product of FrFTs of f and g.*

Note that the convolution of *f* and *g* results in double the length of two individual functions. This is a consequence of how numerical convolution works. **Example 2.** The convolution theorem for constant and delta function has been verified in this example.

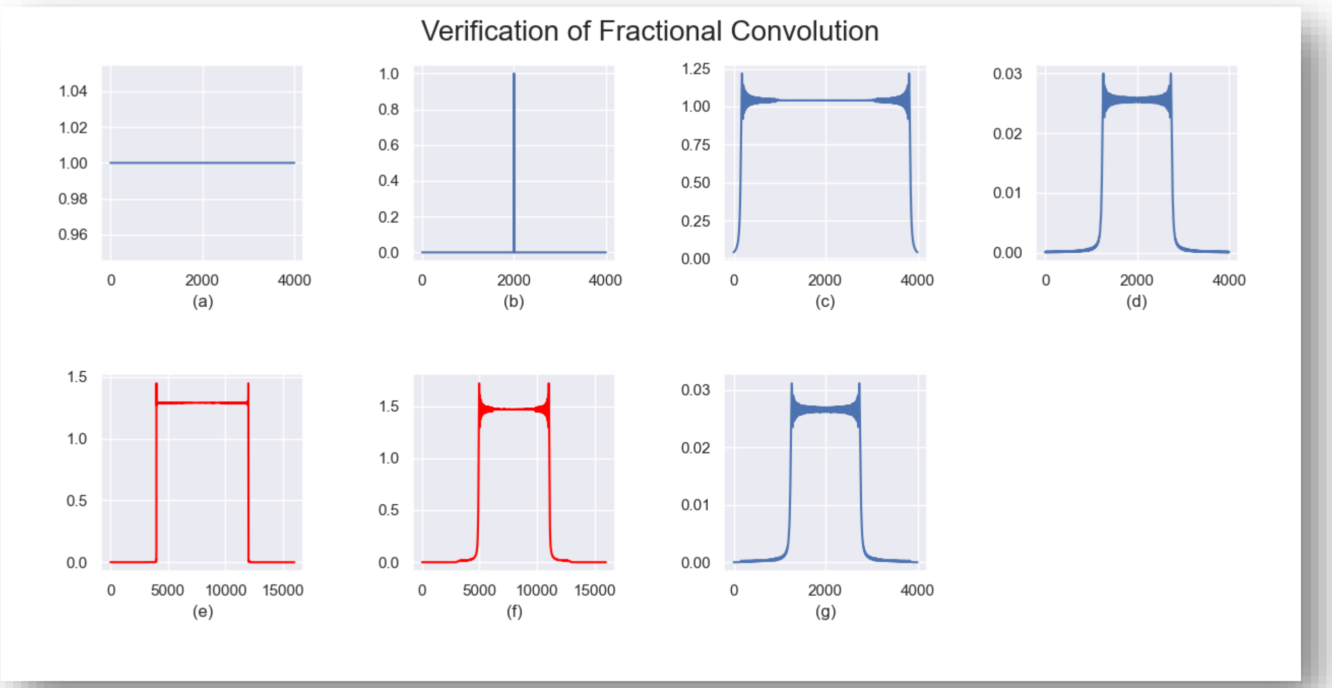

Figure 7: (a) A constant function f. (b) Dirac delta function g, (a) and (b) are input functions for the verification. (c) FrFT of f for a=0.25. (d) FrFT of g for a=0.25. (e) Fractional Convolution h of f and g. (f) FrFT of h at a=0.8. (g) Point wise product of FrFTs of f and g.

3.3 Fractional Convolution in 2D

In this section, we will apply the theories developed in 1D FrFT for 2D practical functions. The main highlight of this section is the convolution theorem in 2D, but to proceed further, we need to define fractional Fourier transform in 2D.

Definition of non-separable two-dimensional fractional Fourier transform exists in many texts; however, the easiest implementation is the employment of a tensor product where we compute 2D FrFT by subsequently employing 1D FrFT to the rows and columns that are present as a grid for input functions.. For example, we will compute 2D FrFT for a 2D discrete rectangular window (fig. 8). The 2D window used in this example is an 8x8 discrete function. $f(x,y) = 1$, for $x, y \in [6, 14]$, otherwise $f(x, y) = 0$. Like 1D FrFT, it is clear that the transform results are varying from the 2D rectangular window to its 2D Fourier transform output ($a=1$, $b=1$). a is the order of FrFT along row transformation, and b is the order along column transformation.

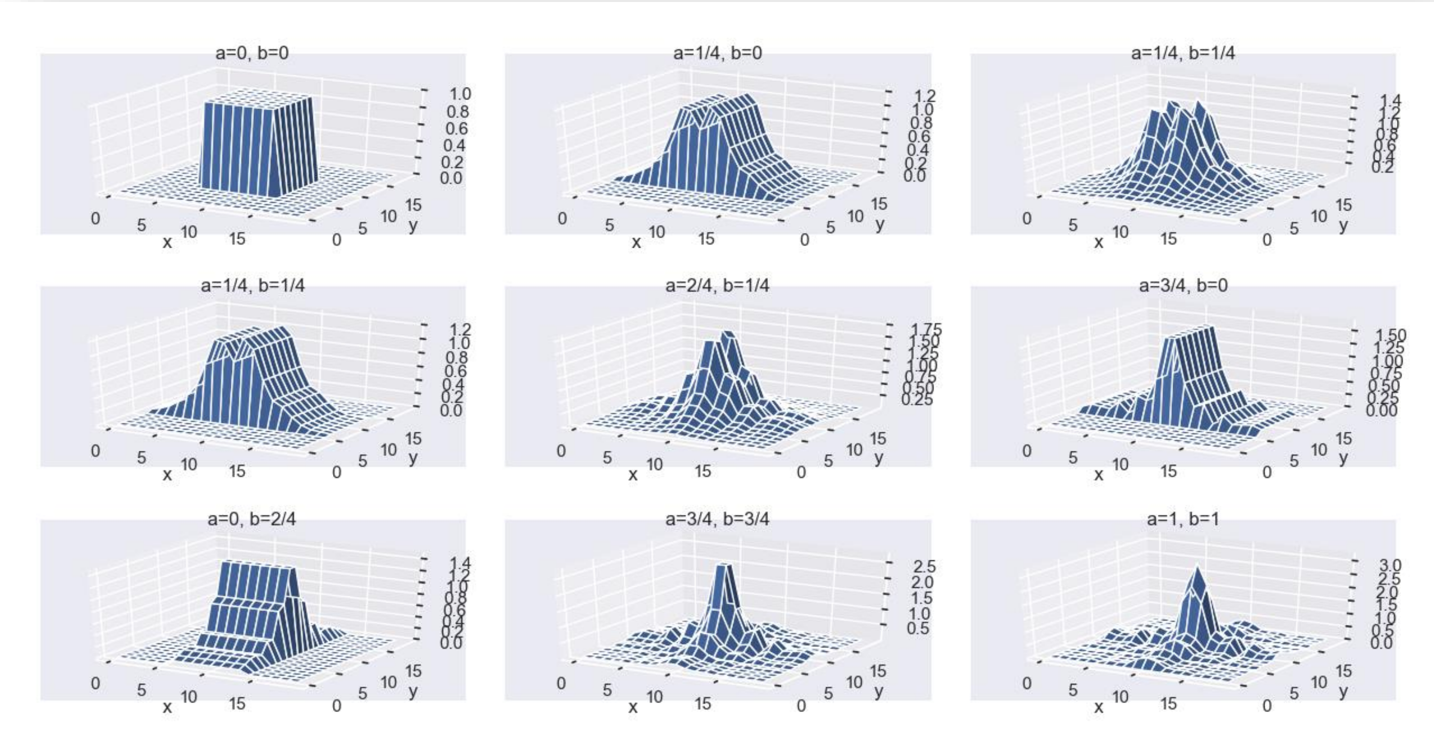

Figure 8: 2D fractional Fourier transform of a rectangular window

Now, the convolution theorem in 2D is readily given here without proof. There is not much difference in the proof from 1D fractional convolution as given in [9].

Definition: *let*

$$\alpha = a * \pi/2, \qquad \beta = b * \pi/2, \qquad \gamma = (\alpha + \beta)/2, \qquad c(\gamma) = (i\cot\gamma)/2$$

$$d(\gamma) = \frac{i\,e^{-i\gamma}}{2\pi \sin\gamma}, \qquad \tilde{f}(x,y) = e^{-c(x^2+y^2)}f(x,y), \qquad \tilde{g} = e^{-c(x^2+y^2)}g(x,y)$$

and define

$$h(x,y) = (f\bigstar g) = d\,e^{c(x^2+y^2)}\left(\tilde{f} * \tilde{g}\right)(x,y), \tag{22}$$

*where * stands for the traditional convolution given by*

$$(f * g)(x,y) = \iint f(x-p, y-q)g(p,q)dpdq$$

Theorem: *Let F, G, and H denote the 2D fractional Fourier transform of f, g, and h respectively and h = (f ★g)*

$$H(u,\ v)\ =\ e^{c(u^2+v^2)}F(u,v)G(u,v) \tag{23}$$

Experimental results

Example 1. A rectangular window and a 2D sine wave are taken as input functions in this example. 2D fractional convolution (23) has been verified

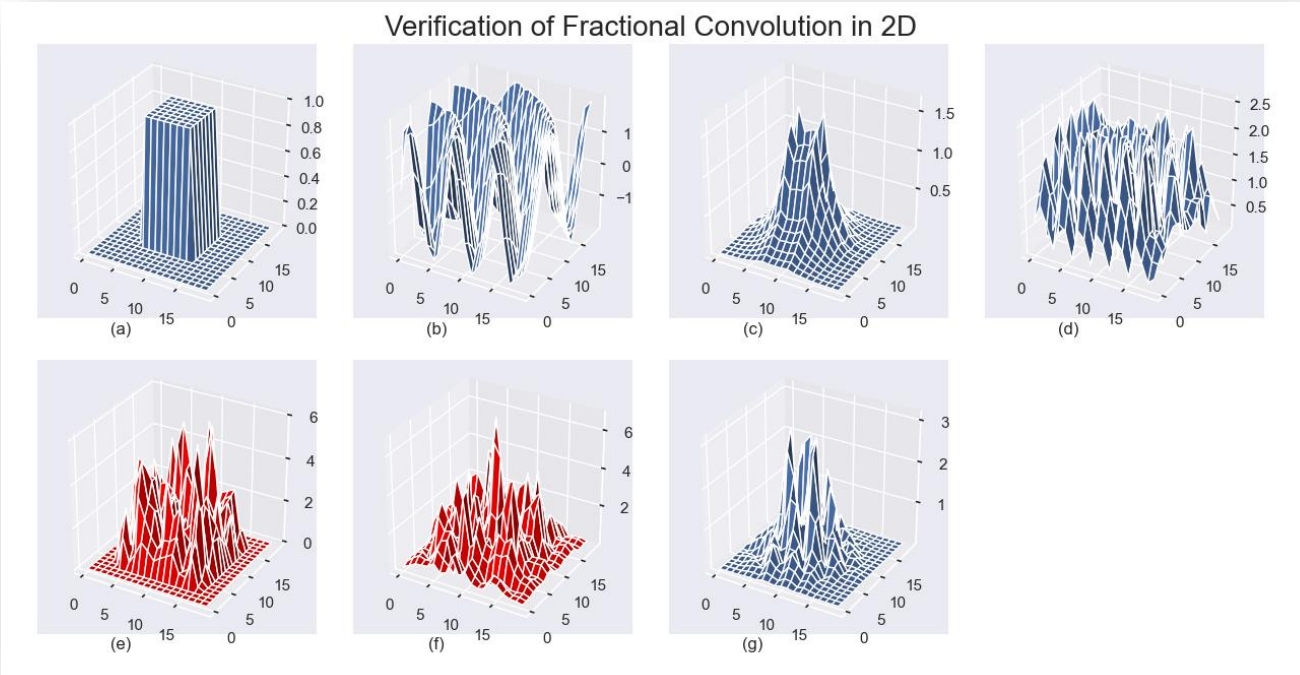

***Figure 9**: (a) A rextangular window f . (b) 2D sine wave g , (a) and (b) are input functions for the verification. (c) FrFT of f for (a=0.2,b=0.3). (d) FrFT of g for (a=0.2,b=0.3). (e) Fractional Convolution h of f and g. (f) FrFT of h at (a=0.2,b=0.3). (g) Point wise product of FrFTs of f and g.*

Plots (f) and (g) represents the expressions in (23)

Example 2. A rectangular window and a 2D sine wave are taken as input functions in this example. 2D fractional convolution (23) has been verified

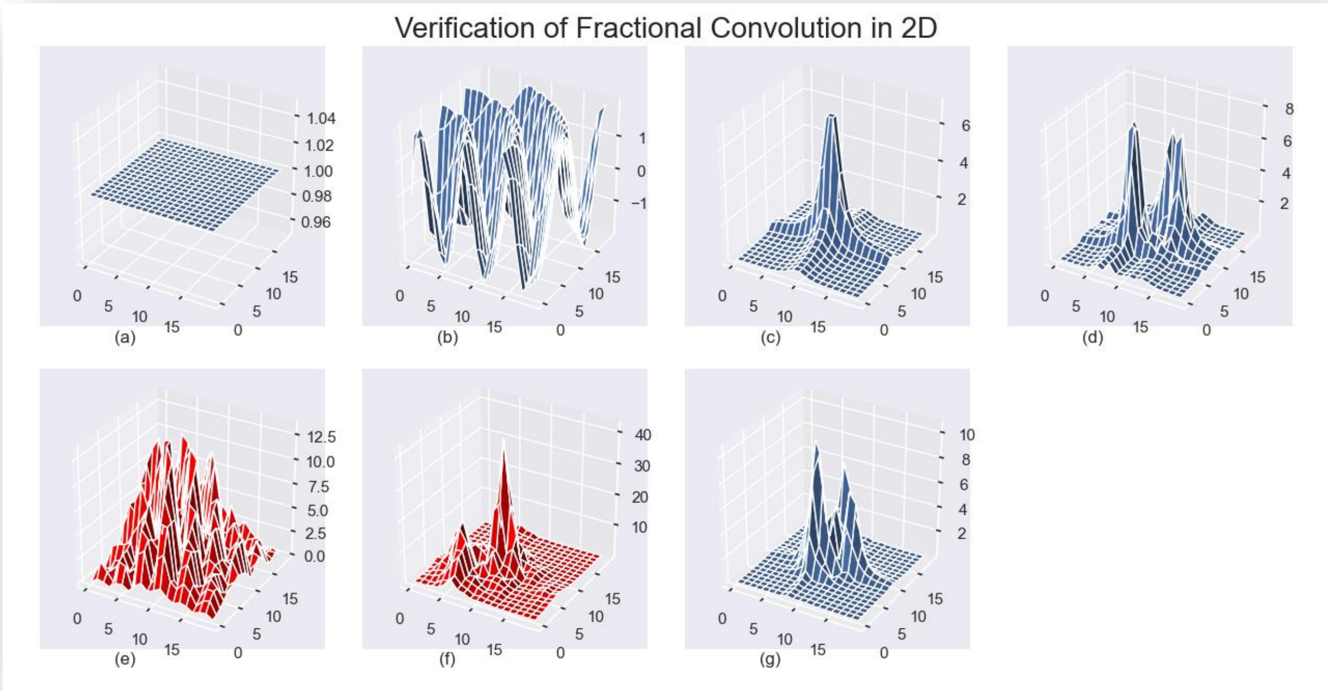

Figure 10: (a) A constant function f. (b) 2D sine wave g, (a) and (b) are input functions for the verification. (c) FrFT of f for (a=0.9,b=0.99). (d) FrFT of g for (a=0.9,b=0.99). (e) Fractional Convolution h of f and g. (f) FrFT of h at (a=0.9,b=0.99). (g) Point wise product of FrFTs of f and g.

Conclusion

We established the concept of fractional Fourier transform representing ordinary Fourier transform as a particular case of FrFT. We studied in detail the various steps involved in the implementation of the fast fractional Fourier transform. We showed the experimental results based on the implementation of the fast fractional Fourier transform. Lastly, we obtained an expression for a new type of convolution for fractional Fourier transform and verified the theorems using graphical analysis. Two-dimensional implementation of fractional convolution was also established and verified.

References

[1] H. M. Ozaktas, B. Barshan, D. Mendlovic, and L. Onural, "Convolution, filtering, and multiplexing in fractional Fourier domains and their relation to chirp and wavelet transforms," J. Opt. Soc. Amer. A, vol. 11, pp. 547-559, 1994

[2] H. M. Ozaktas, O. Arikan, M. A. Kutay and G. Bozdagt, "Digital computation of the fractional Fourier transform," in *IEEE Transactions on Signal Processing*, vol. 44, no. 9, pp. 2141-2150, Sept. 1996.doi: 10.1109/78.536672

[3] Q. Xiang and K. Qin, "The Linear Canonical Transform and Its Application to Time-Frequency Signal Analysis," 2009 International Conference on Information Engineering and Computer Science, Wuhan, China, 2009, pp. 1-4.doi: 10.1109/ICIECS.2009.5365117

[4] L. Cohen, "Time-frequency distributions--A review," Proc. IEEE, vol.77, no. 7, pp 941-981, July 1989

[5] Adhemar Bultheel, Héctor E. Martínez Sulbaran,Computation of the fractional Fourier transform,Applied and Computational Harmonic Analysis,Volume 16, Issue 3,2004,Pages 182-202,ISSN 1063-5203,https://doi.org/10.1016/j.acha.2004.02.001.

[6] L. B. Almeida, "Product and convolution theorems for the fractional Fourier transform," IEEE Trans. Signal Processing Letters, vol. 4, pp. 15–17, 1997.

[7] Zayed, Ahmed. (1998). A convolution and product theorem for the fractional Fourier transform. Signal Processing Letters, IEEE. 5. 101 - 103. 10.1109/97.664179.

[8] Soo-Chang Pei, Min-Hung Yeh, Two dimensional discrete fractional Fourier transform, Signal Processing, Volume 67, Issue 1, 1998, Pages 99-108, ISSN 0165-1684

[9] Ahmed Zayed (2018) Two-dimensional fractional Fourier transform and some of its properties, Integral Transforms and Special Functions, 29:7, 553-570, DOI: 10.1080/10652469.2018.1471689

[10]] L. B. Almeida, "Product and convolution theorems for the fractional Fourier transform," IEEE Trans. Signal Processing Letters, vol. 4, pp. 15–17, 1997.

Appendix (Python code GitHub links)

- Implementation of fractional Fourier transform https://github.com/Farji402/Understanding-Fractional-Fourier-Transform/blob/main/FrFT_code.py
- Convolution theorem for fractional Fourier transform in 1D https://github.com/Farji402/Understanding-Fractional-Fourier-Transform/blob/main/fractional_convolution.py
- Convolution theorem for fractional Fourier transform in 2D https://github.com/Farji402/Understanding-Fractional-Fourier-Transform/blob/main/convolution_2D.py