

Fourier Analysis

Documentation

Paolo Bettelini
Scuola d'Arti e Mestieri di Trevano (SAMT)

Contents

1	Introduction	3
1.1	Abstract	3
1.2	Informations	4
1.3	Scope	4
2	Analysis	5
2.1	Requirements	5
2.1.1	Req-00	5
2.1.2	Req-01	5
2.1.3	Req-02	5
2.1.4	Req-03	6
2.1.5	Req-04	6
2.1.6	Req-05	6
2.2	Ga	7
3	Interactive Boxes	8
3.1	Description	8
3.2	Implementation	8
3.3	List of Functions	9
3.4	Injecting	9
3.5	Example	10
4	Website Structure	11
4.1	Dependency table	11
4.2	Sections	11
4.2.1	Fourier Analysis	11
4.2.2	Requirements	12
4.2.3	Introduction	12
4.2.4	Fourier Series vs Fourier Transform	13
4.2.5	Trigonometric Fourier Series	13
4.2.6	Trigonometric Fourier Series - C term	14
4.2.7	Trigonometric Fourier Series - Coefficients	14
4.2.8	Fourier Series - Conclusion	15
4.2.9	Main ideas - Complex plotting	15
4.2.10	Main ideas - Center of mass	16
4.2.11	Main ideas - Fourier Transform	16
4.2.12	A Simple Example	17

4.2.13	A Simple Example - Coefficients	17
4.2.14	A Simple Example - Conclusion	18
4.2.15	Exponential Fourier Series	18
4.2.16	Fast Fourier Transform	19
4.2.17	Epicyles	19
4.3	Interactive Boxes Implementations	20
4.3.1	Fourier Series 1D	20
4.3.2	Fourier Series 2D	20
4.3.3	Complex Plot	20
4.3.4	Center of mass	20
4.3.5	Fourier Transform	20
5	Conclusion	20
5.1	Further Development	20
5.1.1	Library Design	20
5.1.2	Website Content	20
5.2	Personal Considerations	20
6	Bibliography	20
7	Sitography	20

1 Introduction

1.1 Abstract

Fourier analysis is a method of defining periodic waveforms in terms of trigonometric functions. This branch of mathematics is widely used in signal processing, especially electronics, acoustics and communications. Many notorious algorithms have been developed thanks to Joseph Fourier. Operators such as the Fourier Transform are constantly used in the real world, without these discoveries the world would not be the same. Many software rely in Fourier Analysis, such as for instance Shazam, the famous service for identifying songs. Any audio spectrum visualized processes the signal using Fourier Transform, these are just a few of the many application of this analysis.

1.2 Informations

This is a project of the Scuola Arti e Mestieri di Trevano (SAMT) school under the following circumstances.

- **Section:** Computer Science
- **Year:** Third
- **Class:** Module 306
- **Supervisor:** Luca Muggiasca
- **Title:** Fourier Analysis
- **Start date:** 2021.09.09
- **Deadline:** 2021.12.23

and the following requirements

- **Documentation:** a full documentation of the work done
- **Changelog:** constant changelog for each work session
- **Source code:** working source code of the project

All the source code and documents can be found at <https://github.com/paolobettelini/fourier-series>.
The live version of the final product is available at <https://paolobettelini.github.io/fourier-series>.

1.3 Scope

The scope of this project is to create a website containing various explanations about Fourier Analysis.

2 Analysis

2.1 Requirements

2.1.1 Req-00

Req-00	
Name	Content
Priority	1
Version	2.0
Notes	none
Description	The website must contains a full explanation about Fourier Analysis.

2.1.2 Req-01

Req-01	
Name	Index
Priority	1
Version	2.0
Notes	none
Description	The website must contain an index of all the sections
Subrequirements	
Req-01.0	There must be a section about the topic introduction.
Req-01.1	There must be a section about the knowledge requirements.
Req-01.2	There must be a section about signal processing.
Req-01.3	There must be a section about the Fourier transform.
Req-01.4	There must be a section about the Fourier series.
Req-01.5	There must be a section about how to represent the Fourier series with epicycles.
Req-01.6	There must be a section about Fast Fourier Transform.

2.1.3 Req-02

Req-02	
Name	Responsiveness
Priority	1
Version	1.0
Notes	none
Description	The website must be responsive.

2.1.4 Req-03

Req-03	
Name	Introduction
Priority	1
Version	2.0
Notes	none
Description	The introduction section must contain an interactive Fourier series animation.
Subrequirements	
Req-03.0	The user must be able to draw an arbitrary path.
Req-03.1	The user drawn path is animated with a Fourier series, represented with epicycles.
Req-03.2	The interactive box must contains a timeline slider.
Req-03.3	The interactive box must contain a stop button.
Req-03.4	The interactive box must contain a resume button.

2.1.5 Req-04

Req-04	
Name	Interactiveness
Priority	1
Version	1.0
Notes	none
Description	The website must contain multiple interactive boxes.
Subrequirements	
Req-04.0	All the interactive boxes must follow the design described in Req-03.
Req-04.1	The interactive boxes can contain optional settings.

2.1.6 Req-05

Req-05	
Name	Modularity
Priority	1
Version	1.0
Notes	none
Description	The interactive boxes must share the same base code.

2.2 Ga

3 Interactive Boxes

3.1 Description

InteractiveBoxes is a JavaScript library I wrote for canvas rendering based on the user input. The library injects its content into a HTML div element. The content consists of a canvas element, a stop/resume button and a range slider (the timeline), additional content is injected by the interactive box implementations. The user can interact with the timeline, pause and resume the animation or modify the input by simply drawing onto the canvas.

3.2 Implementation

To create an interactive box you need to create a class that extends `InteractiveBox.js`. The class of your custom interactive box must override some functions, otherwise you will get errors. You will also need to call the super constructor. Here are the declaration of those function in the `InteractiveBox.js` class and its constructor.

```
constructor(name, container, height, width) {  
    ...  
}  
  
draw(ctx) {  
    throw 'The function draw() has not been overwritten'  
}  
  
setPoints(points) {  
    throw 'The function setPoints(points) has not been overwritten'  
}  
  
onTimeTravel(value) {  
    throw 'The function onTimeTravel(value) has not been overwritten'  
}
```

Overriding these functions will produce a class that looks like this

```
class MyCustomBox extends InteractiveBox {  
  
    constructor(name, container, height, width) {  
        super(name, container, height, width)  
  
        // inject extra html, initialize variables, ...  
    }  
  
    draw(ctx) {  
        this.clearCanvas();  
  
        // draw function  
  
        // update timeline  
        this.setTime(...);  
    }  
  
    onTimeTravel(value) {  
        // onTimeTravel function  
    }  
  
    setPoints(points) {  
        // setPoints function  
    }  
}
```


3.3 List of Functions

Here is a list of public functions in `InteractiveBox.js`

Name	Description	Parameters	Returns
<code>constructor()</code>	Constructor	<ul style="list-style-type: none">• name the name of the box• container the div id• height the height of the canvas• width the width of the canvas	void
<code>pause()</code>	Pauses the animation	none	void
<code>resume()</code>	Resumes the animation	none	void
<code>toggle()</code>	Pauses or resumes the animation	none	void
<code>isPlaying()</code>	Returns true if the animation is playing	none	bool
<code>setTime()</code>	Updates the timeline, you should call this in the <code>draw()</code> function	<ul style="list-style-type: none">• value the time value $\in [0; 1]$	void
<code>clearCanvas()</code>	Clears the canvas	none	void
<code>draw()</code>	Called for each frame Must override!	<ul style="list-style-type: none">• ctx The canvas context	void
<code>onTimeTravel()</code>	Called when the user moves the timeline Must override!	<ul style="list-style-type: none">• value the time value $\in [0; 1]$	void
<code>setPoints()</code>	Called when the user draws a path Must override!	<ul style="list-style-type: none">• points array of $\{x,y\}$	void

3.4 Injecting

To inject the interactive box into the site we must create a div element to contain it.

```
<body>
  <!-- Here I place my MyCustomBox-->
  <div id="mycustombox-div">
    </div>
</body>
```

Then, in a JavaScript environment add the box to the div

```
new MyCustomBox('mycustombox1', 'mycustombox-div-box', 500, 500);
```

In order for everything to work you must include the `InteractiveBox.js` file, your `MyCustomBox.js` file and the InteractiveBoxes css stylesheet `boxes.css`.

Note: the name must be unique and the script must be executed after the body has loaded.

3.5 Example

Here is an example of interactive box where the path drawn by the user is progressively drawn on the canvas.

```
class Example extends InteractiveBox {

  #points = []; // The path to be drawn
  #counter = 0; // Drawing process

  constructor(name, container, height, width) {
    super(name, container, height, width)

    this.setPoints(this.#getDefaultPath());
  }

  onTimeTravel(value) {
    // Set counter accoring to value
    this.#counter = value * this.#points.length | 0;
  }

  setPoints(points) {
    this.#counter = 0; // Reset counter
    this.#points = points; // Update points
  };

  draw(ctx) {
    this.clearCanvas(); // Clear the canvas

    // Update counter and update timeline
    this.setTime(this.#counter++ / (this.#points.length - 1));
    if (this.#counter > this.#points.length) {
      this.#counter = 0; // Reset counter
    }

    ctx.beginPath();

    ctx.lineWidth = 2.0;
    ctx.strokeStyle = 'red';

    ctx.moveTo(this.#points[0].x, this.#points[0].y);
    for (var i = 1; i < this.#counter; i++) {
      ctx.lineTo(this.#points[i].x, this.#points[i].y);
    }

    ctx.stroke();
  };

  #getDefaultPath() {
    var circle = [];
    for (var i = 0; i < 100; i++) {
      circle[i] = {
        x: 250 + 50 * Math.cos(Math.PI * 2 / 100 * i),
        y: 250 + 50 * Math.sin(Math.PI * 2 / 100 * i)
      }
    }
    return circle;
  }
}
```

4 Website Structure

4.1 Dependency table

The website relies on various libraries, some of which are not stored locally. This means that the user will query third-party servers, thus the website will not work locally if you do not have a free internet connection.

Dependency table			
Name	Description	Stored	Version
Bootstrap (CSS)	Styling framework	Locally	4.0.0
Bootstrap (JS)	Styling framework	Locally	4.0.0
InteractiveBoxes	Canvas drawing	Locally	1.0
JQuery	Website Manipulation	Locally	3.6.0
Google Fonts	Fonts	Remotely	-
MathJax	LaTeX rendering	Remotely	3.x.x (latest)
Desmos	Graphic calculator	Remotely	1.6

4.2 Sections

The website is made up of several sections, each about a particular topic.

4.2.1 Fourier Analysis

What is Fourier analysis and where is it used.

This section contains the *FourierSeries2D* interactive box.

4.2.2 Requirements

What are the requirements to read the article.

Requirements

This article requires a general good understanding about math. If you want to fully understand every bit of math there are a few requirements:

- Trigonometry
- Calculus
- Complex plane and complex analysis

Even if you don't understand the math involved, the main ideas about Fourier analysis will be explain visually and occasionally with interactive examples.

START

If you understand the following expressions and notations you should have a solid background

$$e^{i\theta} = \cos(\theta) + i \sin(\theta)$$
$$\int_{-\infty}^{\infty} e^{-t^2} dt$$
$$\sum_{n=1}^{\infty} a_n$$

4.2.3 Introduction

Who was Joseph Fourier and what he had discovered.


Introduction

Mathematician and physicist [Joseph Fourier](#) determined that a function can be represented as a series of sines and cosines of different frequencies and different amplitudes.

Fourier is notoriously known for having developed [Fourier series](#) and the [Fourier transform](#), which are the main focus of this article.

We will also cover the [FFT](#) algorithm, which is a fast implementation of the Fourier transform widely used among software.

NEXT - FOURIER SERIES VS TRANSFORM



4.2.4 Fourier Series vs Fourier Transform

What is the difference between the Fourier series and the Fourier transform.

Fourier Series vs Fourier Transform

Fourier Series	Fourier Transform
<p>The Fourier series is the representation of a periodic function with a summation of sine and cosine waves of discrete frequencies. Each wave is weighted according to "how important" it is to represent the original function.</p> <p>Fourier Series are often represented in two ways: trigonometric and exponential. They both work in the same way, but the exponential one is also defined on the complex plane and as we'll see, has a nicer, more elegant form.</p>	<p>The Fourier transform is an operation that transforms a signal from time-domain to a continuous frequency-domain. The function can be a generic, not necessarily periodic function $f(x)$. The output of the Fourier transform \mathcal{F} is a complex-valued function whose absolute value represents the magnitude of each frequency.</p> $\mathcal{F}\{f(t)\} = \hat{f}(\xi)$

[NEXT - TRIGONOMETRIC FOURIER SERIES](#)

4.2.5 Trigonometric Fourier Series

Representing a periodic function using a sum of trigonometric functions.

Trigonometric Fourier Series

A function $f(t)$ is periodic if there is a positive number T (the period of f) such that

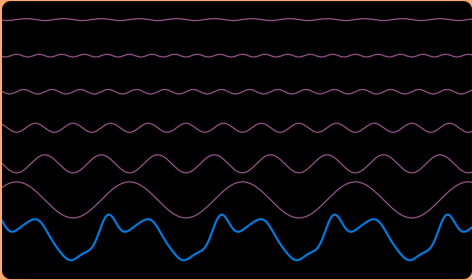
$$f(t + nT) = f(t) \quad \forall t \in D_f, n \in \mathbb{Z}$$

We can represent a periodic function using a sum of sines and cosines, for each discrete frequency we have a wave with its own weight (its amplitude).

$$f(t) = C + \sum_{n=1}^{\infty} a_n \cos\left(\frac{2\pi n t}{T}\right) + b_n \sin\left(\frac{2\pi n t}{T}\right)$$

We're computing a sum from $n = 1$ to infinity, where n represents each discrete frequency. The term $\frac{2\pi n}{T}$ controls the frequency based on n . A normal sine or cosine wave oscillates every 2π , with this modification e.g. $n = 5$ means that the function will oscillate 5 times within that span. The terms a_n and b_n control how much that particular frequency is important. You might notice that with this we can only represent functions "lying" on the x -axis. To resolve this problem we add a generic constant term C to shift the function up/down.

>> Note: We only need the discrete frequencies (1 Hz, 2 Hz, ...) to represent the function, although the Fourier transform gives us a continuous frequency analysis.



Sum of waves

[NEXT - C TERM](#)

4.2.6 Trigonometric Fourier Series - C term

Finding the C term.

Trigonometric Fourier Series - C term

Great! One small problem, what are a_n , b_n and C ? Here we manipulate the Fourier series definition to find these values given $f(t)$ on a period $[t_0; t_0 + T]$.

>> **Note:** To simplify the equations I'm going to define $w_k = \frac{2\pi k}{T}$

Starting from the C term, we take the integral over the generic period $[t_0; t_0 + T]$ on both sides

$$\int_{t_0}^{t_0+T} f(t) dt = \int_{t_0}^{t_0+T} h dt + \sum_{n=1}^{\infty} \left[a_n \int_{t_0}^{t_0+T} \cos(w_n t) dt + b_n \int_{t_0}^{t_0+T} \sin(w_n t) dt \right]$$

If you think about it, the integral over a full period of a function such as $\sin(x)$ or $\cos(x)$ is 0. If we consider $\sin(w_n x)$ or $\cos(w_n x)$ the function will make more full cycles in the span of the interval T , all of which yield an area of 0.

$$\begin{aligned} \int_{t_0}^{t_0+T} f(t) dt &= \int_{t_0}^{t_0+T} C dt + \sum_{n=1}^{\infty} a_n \cdot 0 + b_n \cdot 0 \\ &= \int_{t_0}^{t_0+T} C dt \\ &= C \int_{t_0}^{t_0+T} dt \\ &= C[x]_{t_0}^{t_0+T} \\ &= C(t_0 + T - t_0) \\ &= CT \end{aligned}$$

concluding that

$$C = \frac{1}{T} \int_{t_0}^{t_0+T} f(t) dt$$

NEXT - COEFFICIENTS

4.2.7 Trigonometric Fourier Series - Coefficients

Finding the coefficients a_n and b_n .

Trigonometric Fourier Series - Coefficients

Finding a_n

Again, we take the integral over the period $[t_0; t_0 + T]$ on both sides, but first we multiply everything by $\cos(w_k t)$

$$\int_{t_0}^{t_0+T} f(t) \cos(w_k t) dt = h \int_{t_0}^{t_0+T} \cos(w_k t) dt + \sum_{n=1}^{\infty} \left[a_n \int_{t_0}^{t_0+T} \cos(w_n t) \cos(w_k t) dt + b_n \int_{t_0}^{t_0+T} \sin(w_n t) \cos(w_k t) dt \right]$$

By using orthogonality relationships or by literally evaluating the above integrals, we get the following

$$\begin{aligned} \int_{t_0}^{t_0+T} f(t) \cos(w_k t) dt &= \int_{t_0}^{t_0+T} a_k \cos^2(w_k t) dt \\ &= a_k \left(\frac{T}{2} \right) \end{aligned}$$

concluding that

$$a_k = \frac{2}{T} \int_{t_0}^{t_0+T} f(t) \cos(w_k t) dt$$

Finding b_n

To find b_n , we do the exact same thing, but instead of multiplying by $\cos(w_n t)$, we multiply by $\sin(w_n t)$, ending up with

$$b_k = \frac{2}{T} \int_{t_0}^{t_0+T} f(t) \sin(w_k t) dt$$

>> **Note:** $C = \frac{a_0}{2}$

NEXT - CONCLUSION

4.2.8 Fourier Series - Conclusion

Conclusion on the last chapters.

Fourier Series - Conclusion

Under appropriate conditions, we can represent a periodic function $f(t)$ on an interval $t_0; t_0 + T$ by

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(w_n t) + b_n \sin(w_n t)$$
$$a_k = \frac{2}{T} \int_{t_0}^{t_0+T} f(t) \cos(w_k t) dt$$
$$b_k = \frac{2}{T} \int_{t_0}^{t_0+T} f(t) \sin(w_k t) dt$$

where

$$w_k = \frac{2\pi k}{T}$$

NEXT - COMPLEX PLOTTING

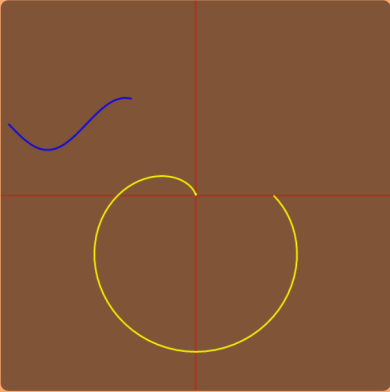
4.2.9 Main ideas - Complex plotting

Plotting a function around the origin in the complex plane using Euler's identity.

Main ideas - Complex plotting

To start we will plot our signal in the complex plane. We will plot it while making it rotate around the origin $0 + 0i$. Recall that [Euler's Formula](#) tells us that the function e^{it} is a rotation around the origin, also called the unit circle. If we multiply our signal by this circle, it will follow its path, achieving a polar plot-like graph: $f(t)e^{it}$. Now, the rotational function makes a full cycle every 2π , we can plot our signal at a different speed (different frequencies). To do so we will multiply the time of the rotational function by $2\pi\xi$ where ξ is the frequency. We add the 2π term so that if your frequency is 1, we will have a rotation each second rather than every 2π seconds (~ 6.28 s). Furthermore, we want the unit circle to rotate clockwise instead of counter-clockwise. We will just add the negative sign to the time t argument of the rotational function. Our final function for now is $f(t)e^{-2\pi i t}$. You can see the animation next to this paragraph, you can vary the frequency or even draw your own signal.

NEXT - CENTER OF MASS



The image shows a complex plane with a horizontal real axis and a vertical imaginary axis. A blue curve representing a signal $f(t)$ is plotted in the upper-left quadrant. A yellow unit circle is centered at the origin. A red line segment connects the tip of the blue curve to the origin, passing through the yellow circle. Below the plot, there is a control panel with a play button, a frequency slider set to 1, and buttons for 'sin' and 'cos'.

Frequency: 1

sin cos

4.2.10 Main ideas - Center of mass

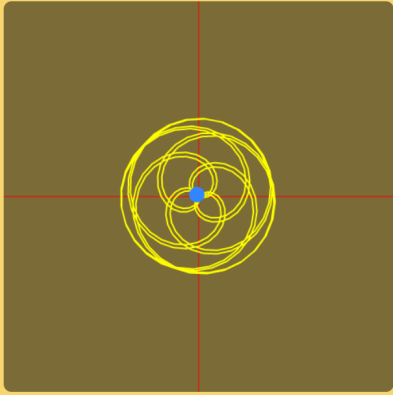
Computing the center of mass of $f(t)e^{-2\pi i t \xi}$

Main ideas - Center of mass

The next step is to create a new function. Instead of plotting our signal as time increases with a given frequency ξ , we plot all of the signal at once but the frequency changes over time. Here we are moving from a time-dependant function to a frequency-dependant function. The argument is no longer the time $f(t)$ but rather the frequency with which we want to plot our signal around the origin $f(\xi)$. Now, what is the center of mass? The center of mass is basically the average point of the function, which is a complex number. You might notice that the center of mass (the blue dot) in the animation changes as the frequency changes. To find its value for a certain frequency, we can sample a bunch of points from the function and then divide it by the number of samples. This approach works when we are dealing with discrete functions, such as the animation (the signal you draw is a set of points), but we would need an infinite amount of samples when the signal is continuous. An infinite amount of precision is achievable using an integral. To find the center of mass when the frequency is ξ we integrate our complex function over a certain period of time, and then divide it by the time length.

$$\frac{1}{t_1 - t_0} \int_{t_0}^{t_1} f(t) e^{-2\pi i t \xi} dt$$

NEXT - FOURIER TRANSFORM



II

sine

4.2.11 Main ideas - Fourier Transform

What is the Fourier transform operator.

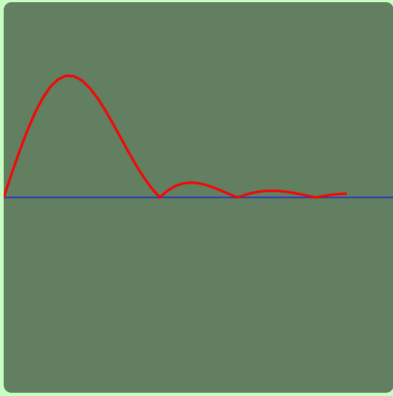
Main ideas - Fourier Transform

The next animation is distance of the center of mass from the origin as the frequency changes. The key concept is that when the frequency matches the period, the center of mass is unusually further from the origin. Go back to the last section and look at the blue dot moving far away from the origin when the frequency is the right one (reset the sine or cosine wave so that it is clearer). When the frequency is a component of the function, this distance peaks. If the function is composed of multiple frequencies, the same distance will peak multiple times, and more it peaks, the more the frequency is present in the original function.

The function that represent the center of mass as the frequency changes is called Fourier transform. Actually not quite, the Fourier transform is defined with the integral over \mathbb{R} and is not divided by the time span. The absolute value (distance from the origin) of this function if the amount of all continuous frequencies present in the original signal.

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(t) e^{-2\pi i t \xi} dt$$

NEXT - AN EXAMPLE



II

sine(1x) sine(2x) sine(3x)

4.2.12 A Simple Example

Computing the Fourier series of a simple function.

A Simple Example

Let's look at a simple example. We are going to derive the Fourier series of a function $f(x)$ defined as such:

$$f(x) = \begin{cases} -1 & \text{if } 0 < x < \pi \\ +1 & \text{if } -\pi < x < 0 \end{cases}$$

The period of this function is $T = 2\pi$. We can already simplify the $\frac{2\pi}{T}$ term, leaving us with

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(nx) + b_n \sin(nx)$$

NEXT - COEFFICIENTS

4.2.13 A Simple Example - Coefficients

Finding the coefficients of the Fourier series.

A Simple Example - Coefficients

Finding a_n

First, we need to find a_n . Simplifying $\frac{2\pi}{T}$ and $\frac{T}{2}$ we get

$$a_n = \int_{-\pi}^{\pi} f(x) \cos(nx) dx$$

Looking at the graph we notice that we can split the integral into two parts at $x = 0$. On the left part, the function is $-\cos(nx)$, while on the right part the function is $\cos(nx)$.

$$\begin{aligned} a_n &= \frac{1}{\pi} \int_{-\pi}^0 -\cos(nx) dx + \frac{1}{\pi} \int_0^{\pi} \cos(nx) dx \\ &= -\frac{1}{\pi} \int_{-\pi}^0 \cos(nx) dx + \frac{1}{\pi} \int_0^{\pi} \cos(nx) dx \\ &= -\frac{1}{\pi} \left[\frac{\sin(nx)}{n} \right]_{-\pi}^0 + \frac{1}{\pi} \left[\frac{\sin(nx)}{n} \right]_0^{\pi} \\ &= -\frac{1}{\pi} \left[\frac{\sin(\pi n)}{n} \right] + \frac{1}{\pi} \left[\frac{\sin(\pi n)}{n} \right] \\ &= \left(\frac{1}{\pi} - \frac{1}{\pi} \right) \left[\frac{\sin(\pi n)}{n} \right] \\ &= 0 \end{aligned}$$

a_n is always going to be 0. We can remove the $a_n \cos(nx)$ and $\frac{a_0}{2}$ terms from the series.

NEXT - CONCLUSION

Finding b_n

Now the same thing for b_n

$$b_n = \int_{-\pi}^{\pi} f(x) \sin(nx) dx$$

Again, we split the integral into two parts

$$\begin{aligned} b_n &= -\frac{1}{\pi} \int_{-\pi}^0 \sin(nx) dx + \frac{1}{\pi} \int_0^{\pi} \sin(nx) dx \\ &= -\frac{1}{\pi} \left[-\frac{\cos(nx)}{n} \right]_{-\pi}^0 + \frac{1}{\pi} \left[-\frac{\cos(nx)}{n} \right]_0^{\pi} \\ &= -\frac{1}{\pi} \left[\frac{1}{n} + \frac{\cos(\pi n)}{n} \right] + \frac{1}{\pi} \left[\frac{-\cos(\pi n)}{n} + \frac{1}{n} \right] \\ &= -\frac{1}{\pi} \left[\frac{\cos(\pi n) - 1}{n} \right] + \frac{1}{\pi} \left[\frac{1 - \cos(\pi n)}{n} \right] \\ &= \frac{2}{\pi} \cdot \frac{1 - \cos(\pi n)}{n} \\ &= \frac{2 - 2 \cos(\pi n)}{\pi n} \end{aligned}$$

4.2.14 A Simple Example - Conclusion

Demonstrating the Fourier series by plotting it.

A Simple Example - Conclusion

Given b_n our series is now complete!

$$f(x) = \sum_{n=1}^{\infty} \frac{2 - 2 \cos(\pi n)}{\pi n} \cdot \sin(\pi x)$$

We won't simplify this further, therefore this is our final result.

The effort pays off when we graph this function, as more terms are added, the function looks more and more like the original square wave. You can drag the slider by opening the left panel.

>> Note: If $f(x)$ is even, the coefficient b_n will always be equal to zero. If $f(x)$ is odd, a_n will always be equal to zero.

[NEXT - EXPONENTIAL FORM](#)

4.2.15 Exponential Fourier Series

Defining the Fourier series using Euler's Identity.

Exponential Fourier Series

The exponential Fourier Series is the same thing but extended to the complex plane, we can use Euler's identity to manipulate the real Fourier series and get the following expression

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \frac{1}{2} (a_n - ib_n) e^{i\omega_n t} + \frac{1}{2} (a_n + ib_n) e^{-i\omega_n t}$$

Now, let n be also negative. With an appropriate coefficient c_n we can arrange the series as

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{i\omega_n t}$$

The coefficient c_n can be computed as

$$c_n = \frac{1}{T} \int_{t_0}^{t_0+T} f(t) e^{-i\omega_n t} dt$$

[NEXT - FFT](#)

4.2.16 Fast Fourier Transform

What is the Fast Fourier Transform algorithm.

Fast Fourier Transform

The [Fast Fourier Transform](#) is a computer algorithm to compute the [Discrete Fourier Transform](#). The fastest and most used implementation of this algorithm is FFTw, a C subroutine library written at MIT.

When computing the DFT on a (discrete) signal $f(t)$, we take the average of all the points of $f(t)e^{-2\pi i\xi}$. This process is repeated for each frequency ξ . Computing all of these values is a $O(N^2)$ operation, but with the FFT algorithm we can decrease the number of operations to $O(N \log(N))$. Many FFT algorithms depend on that fact that $e^{-2\pi i/N}$ is a [root of unity](#).

There are plenty of FFT algorithms, here's a few: [Cooley-Tukey FFT](#), [Prime-factor FFT](#), [Bruun's FFT](#), [Rader's FFT](#), [Chirp Z-transform](#), [Hexagonal fast Fourier transform](#)

There is also another version called SFT (Sparse Fourier Transform), which is a DFT for handling big data signals, and is mainly used in GPS synchronization.

NEXT - EPICYCLES

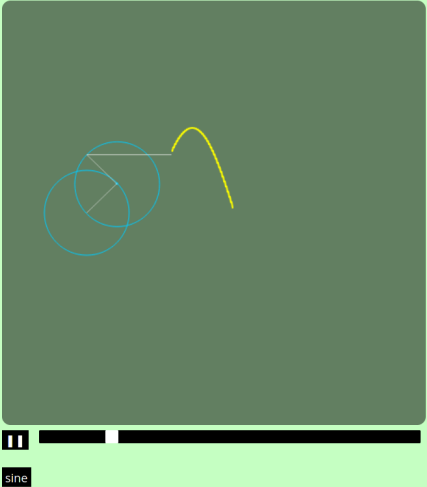
4.2.17 Epicycles

How the animation in Chapter. 1 works.

Epicycles

So how does the cool epicycles animation work? First of all we need apply the Fourier transform operation, however the drawing is just a set of points, it's a discrete function rather than a continuous one, this means that we'll need to use the [Discrete Fourier Transform](#) operator. Each circle represents a discrete frequency, and each center revolves around the previous circle's circumference. The radius of the circle is the magnitude of the current frequency, which is the absolute value of the Fourier transform at that frequency. The revolution is based on the time passed and the phase of the Fourier transform.

THANK YOU



4.3 Interactive Boxes Implementations

4.3.1 Fourier Series 1D

4.3.2 Fourier Series 2D

4.3.3 Complex Plot

4.3.4 Center of mass

4.3.5 Fourier Transform

5 Conclusion

5.1 Further Development

5.1.1 Library Design

Something that could be improved is OOP hierarchy. I could make two classes extending `InteractiveBox.js`: `InteractiveBox1D.js` and `InteractiveBox2D.js`. This way every implementation of the library extends either one of these two classes. The first one lets only the user draw a one-dimensional signal. This is a problem in the current version, since some interactive boxes implementation only processes the y-coordinate of the user drawn path, without blocking him from drawing a two-dimensional path.

5.1.2 Website Content

The website lacks of an explanation about the Fast Fourier Transform and how to implement it. Another covered topic could be the inverse Fourier operators.

5.2 Personal Considerations

I'm really happy with how the websited turned out and I've gained deep knowledge about Joseph Fourier's work. However, I am dissatisfied with what I have written. There are so many topics concerning Fourier analysis and I wish I could have covered more.

I think I managed the timing well even though I finished later than expected.

The nature of this project is very creative. At the beginning I didn't have a precise picture of what I was going to put on the website, I chose the content as I was studying the topic, but in the end I managed to almost respect my initial idea.

6 Bibliography

7 Sitography