Fourier Analysis
Documentation

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

Contents

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	1.0
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.
Req-03_5	The interactive box must contain a slider for the animation speed.

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	All 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 it's content into a HTML div element. The content consists of a canvas element, a stop/resume button and a range slder (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
constructor()	Constructor	• **name** the name of the box • **container** the div id • **height** the height of the canvas • **width** the width of the canvas	void
pause()	Pauses the animation	none	void
resume()	Resumes the animation	none	void
toggle()	Pauses or resumes the animation	none	void
isPlaying()	Returns `true` if the animation is playing	none	bool
setTime()	Updates the timeline, you should call this in the draw() function	• **value** the time value $\in [0; 1]$	void
clearCanvas()	Clears the canvas	none	void
draw()	Called for each frame **Must override!**	• **ctx** The canvas context	void
onTimeTravel()	Called when the user moves the timeline **Must override!**	• **value** the time value $\in [0; 1]$	void
setPoints()	Called when the user draws a path **Must override!**	• **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

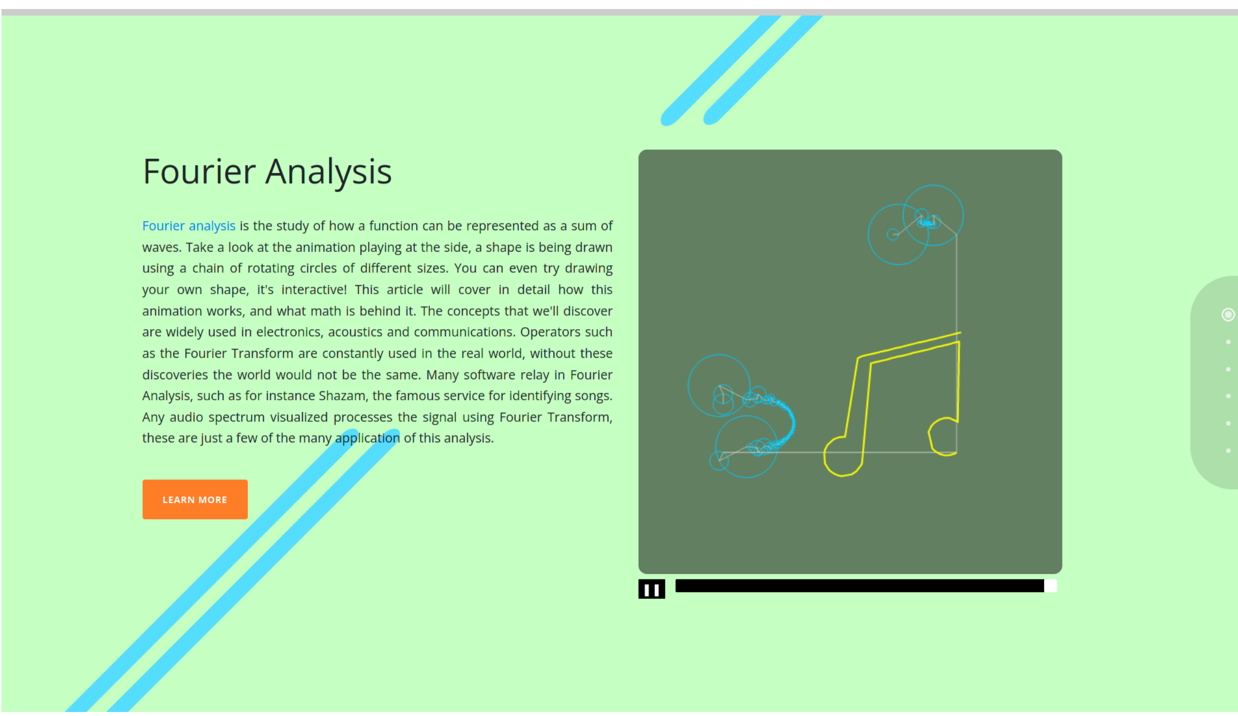
4.2 Sections

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

4.2.1 Fourier Analysis

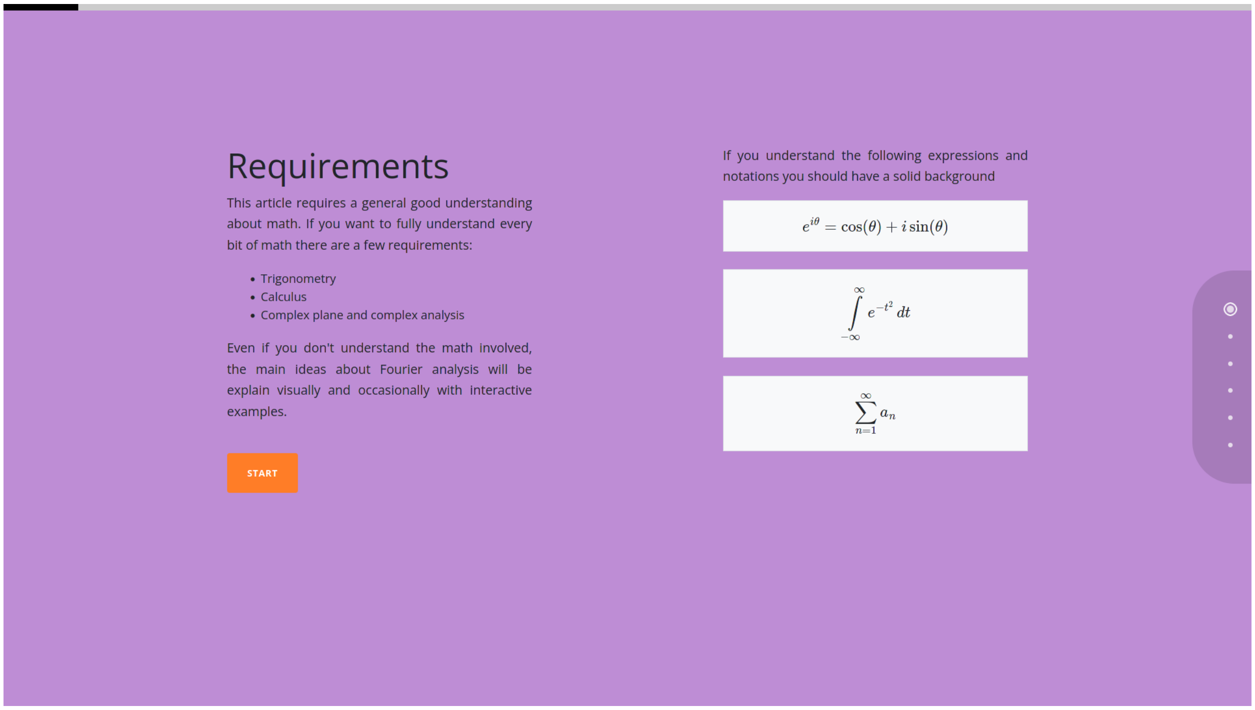
What is Fourier analysis and where is it used.

This section contains the *FourierSeries2D* interactive box.



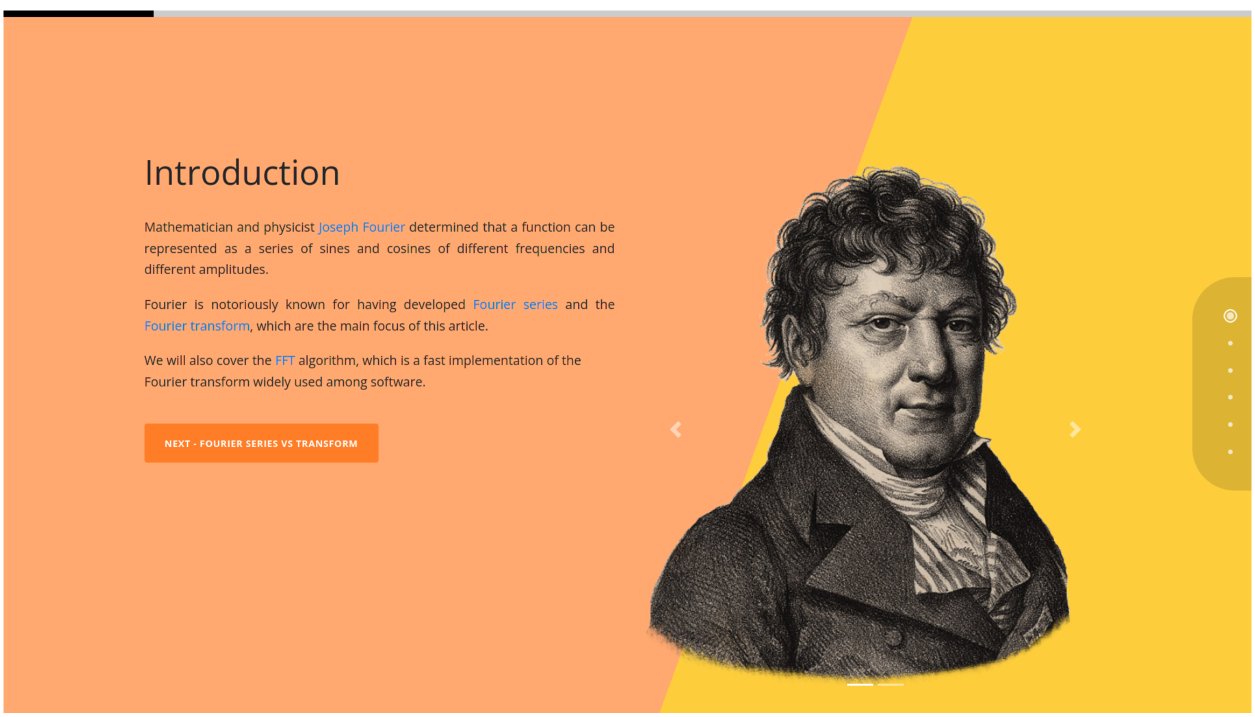
4.2.2 Requirements

What are the requirements to read the article.



4.2.3 Introduction

Who was Joseph Fourier and what he had discovered.



4.2.4 Fourier Series vs Fourier Transform

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

Fourier Series vs Fourier Transform

Fourier Series

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 orifinal function.

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.

NEXT - TRIGONOMETRIC FOURIER SERIES

Fourier Transform

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 period 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.

$$\mathcal{F}\{f(t)\} = \hat{f}(\xi)$$

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

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

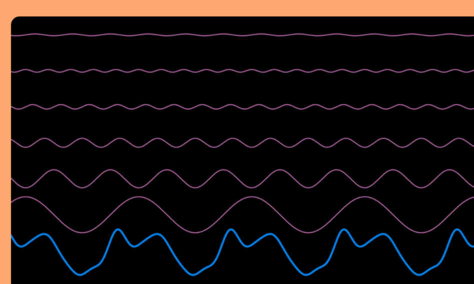
We can represent a periodic function using a sum of sines and cosines, for each discrete frequency we have a wave with its own weight (its amplitude).

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

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 of 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_b and b_n control how much that particular frequency is important. You might notice that with this we can only represent functions "laying" 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, althought the Fourier transform gives us a continuous frequency analysis.

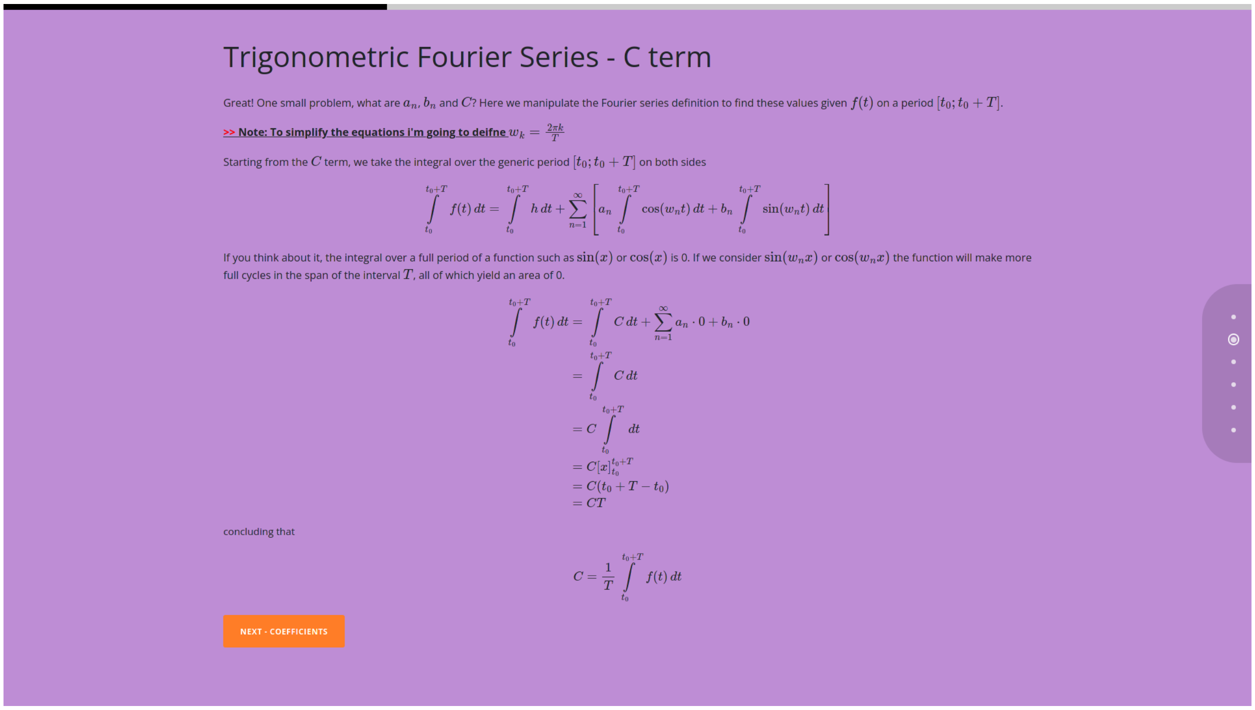
NEXT - C TERM



Sum of waves

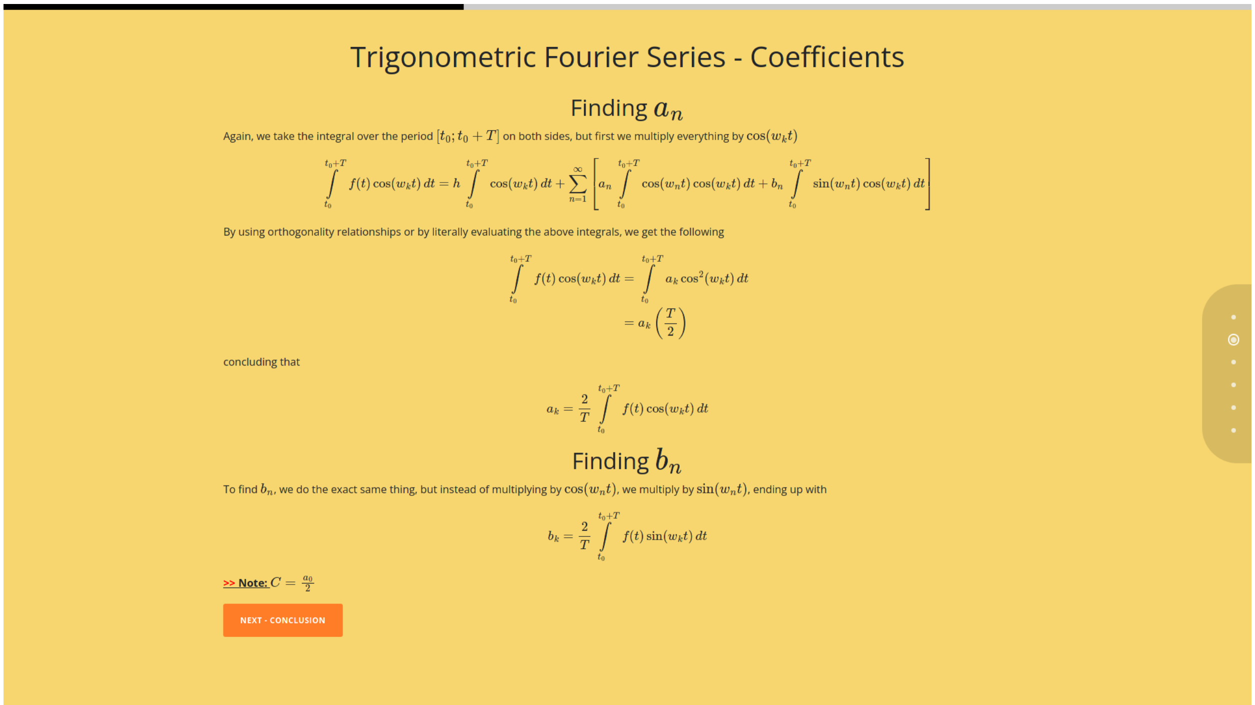
4.2.6 Trigonometric Fourier Series - C term

Finding the C term.



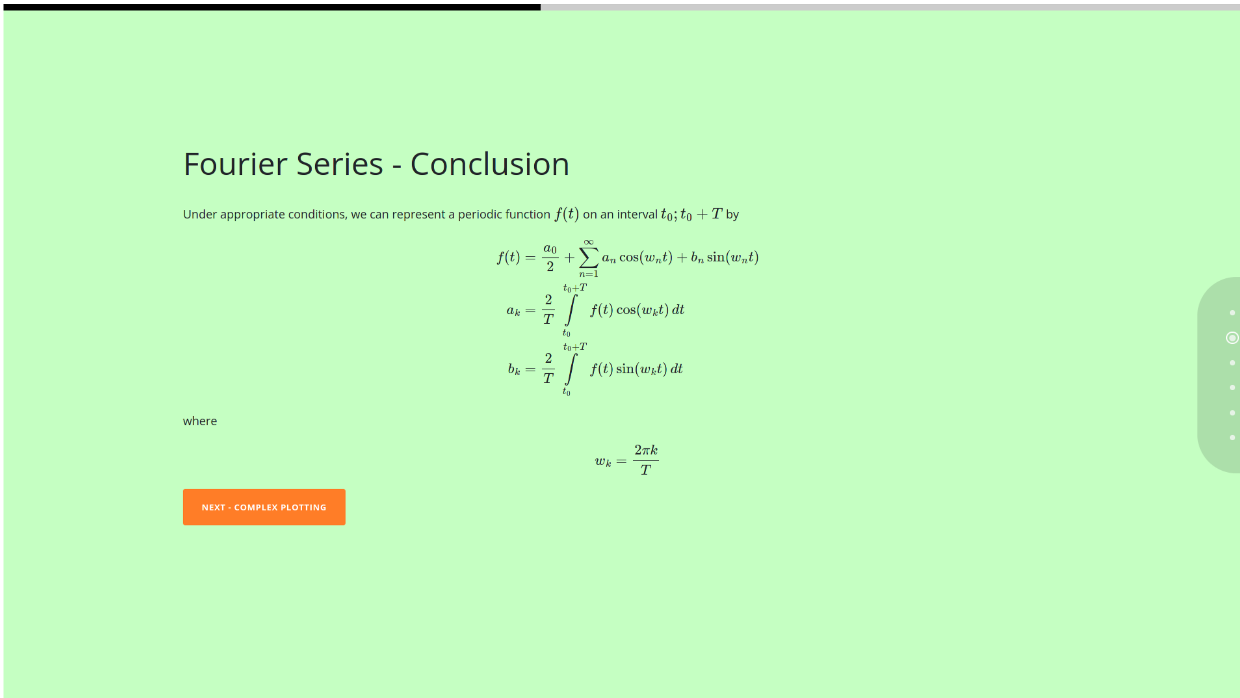
4.2.7 Trigonometric Fourier Series - Coefficients

Finding the coefficients a_n and b_n.



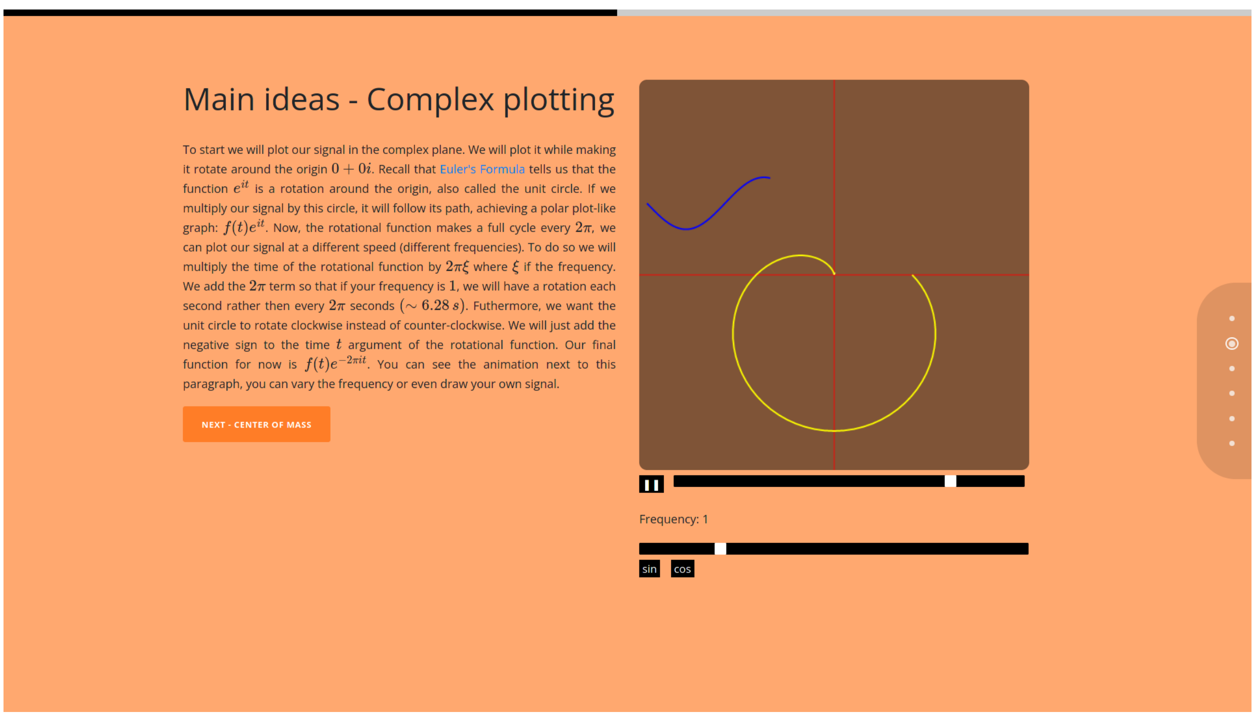
4.2.8 Fourier Series - Conclusion

Conclusion on the last chapters.



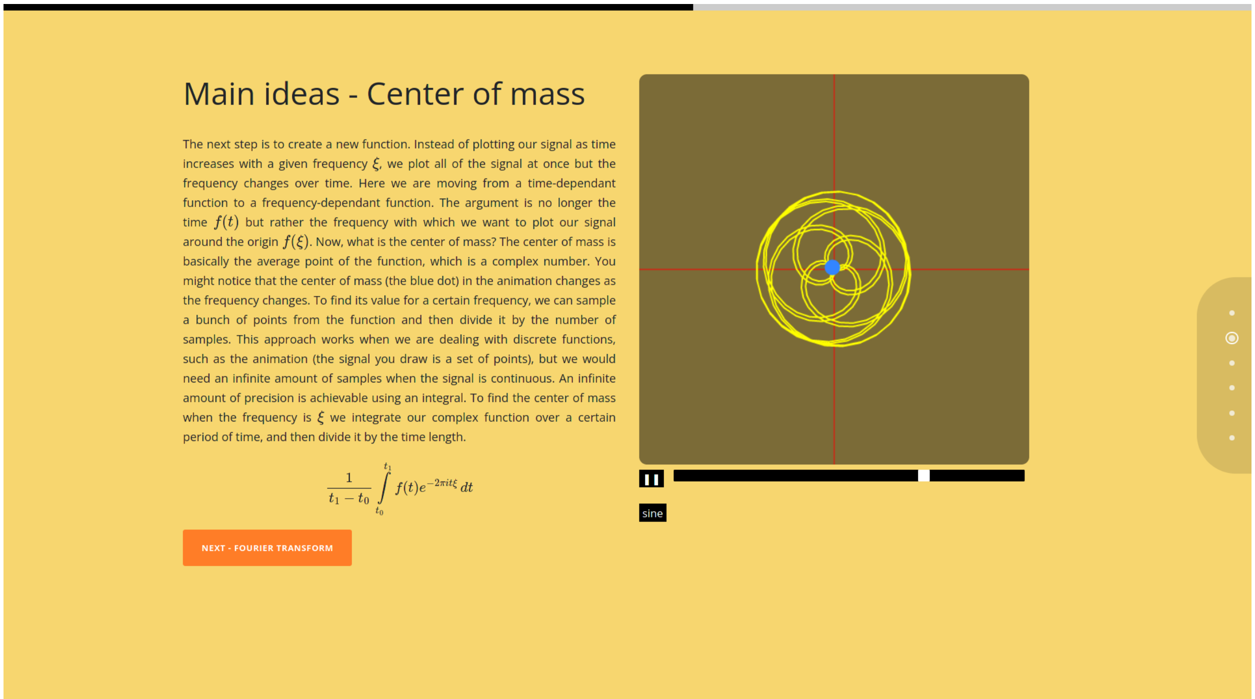
4.2.9 Main ideas - Complex plotting

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



4.2.10 Main ideas - Center of mass

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



4.2.11 Main ideas - Fourier Transform

What is the Fourier transform operator.

4.2.12 A Simple Example

Computing the Fourier series of a simple function.

A Simple Example

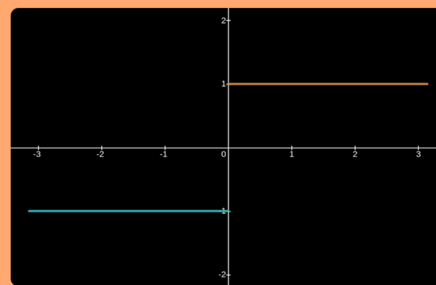
Let's look at a simple example. We are going to derive the Fourier series of a function $f(x)$ defined as such:

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

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 $\cos(nx)$.

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

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

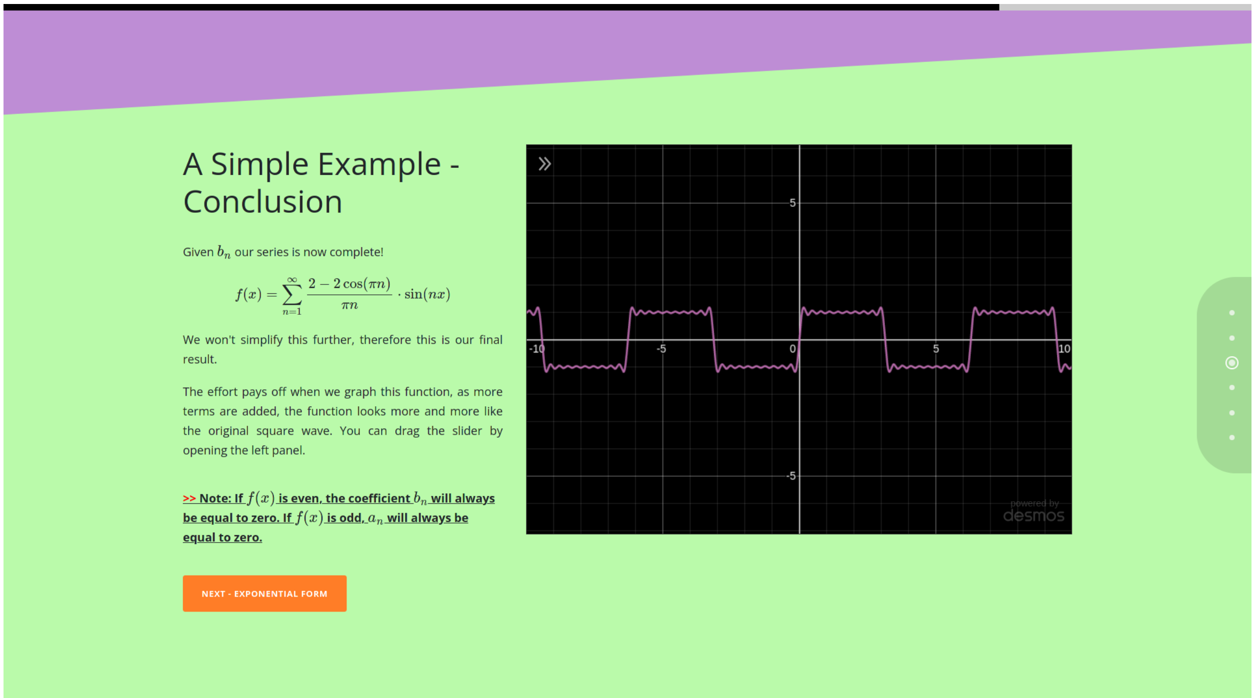
$$b_n = \int_{-\pi}^{\pi} f(x)\sin(nx)\,dx$$

Again, we split the integral into two parts

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

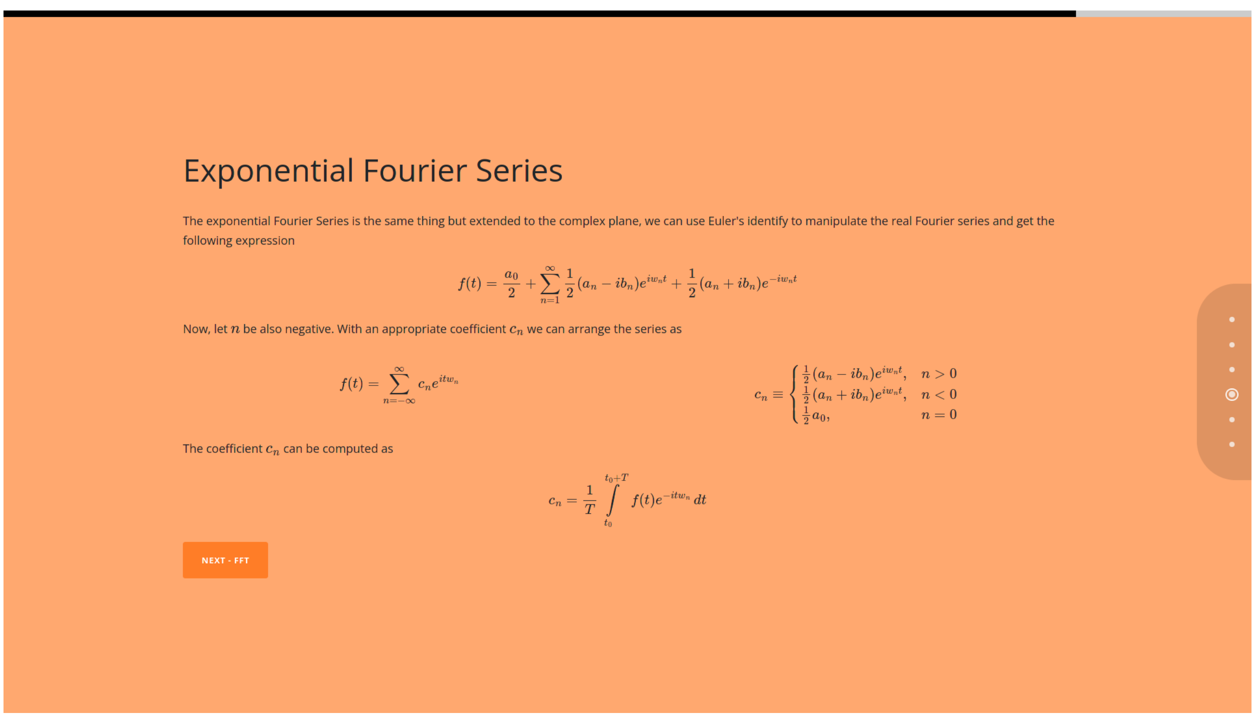
4.2.14 A Simple Example - Conclusion

Demostrating the Fourier series by plotting it.



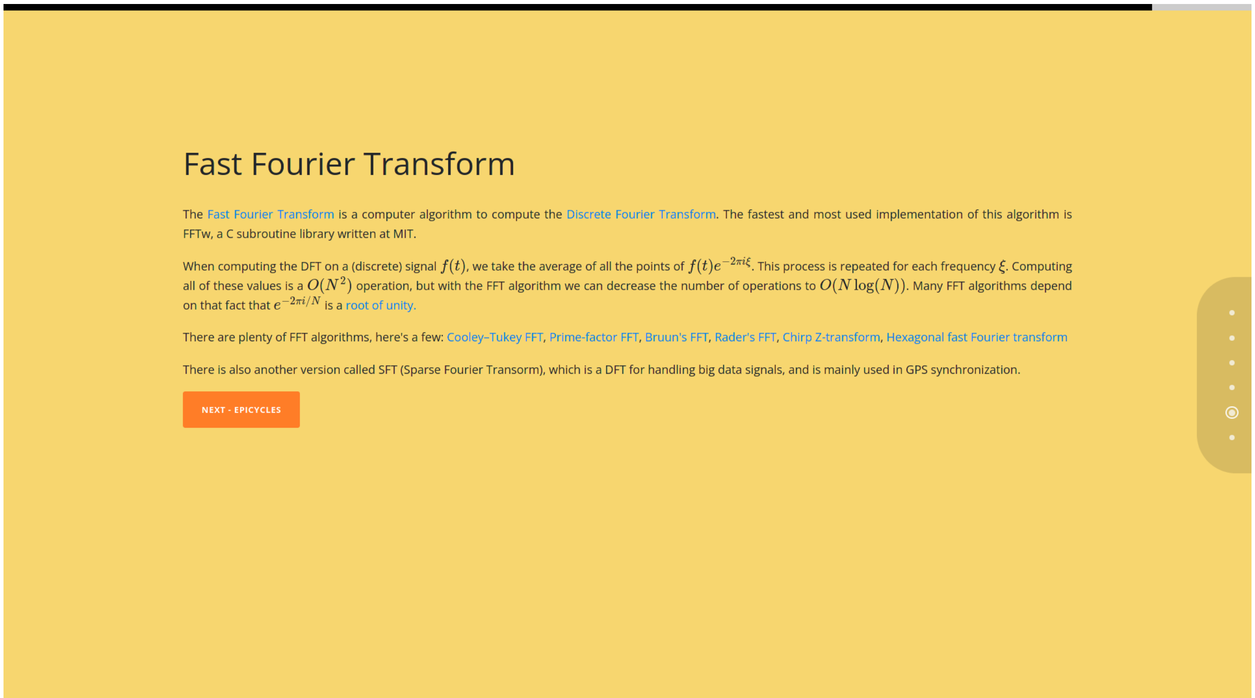
4.2.15 Exponential Fourier Series

Defining the Fourier series using Euler's Identity.



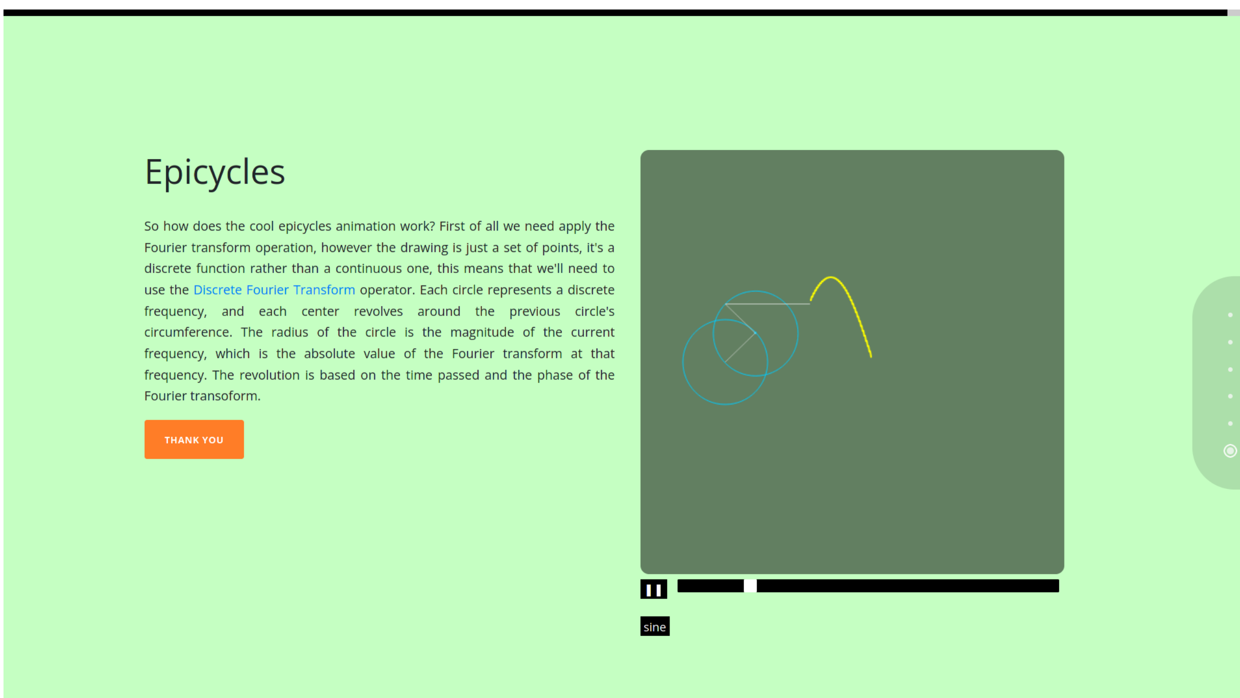
4.2.16 Fast Fourier Transform

What is the Fast Fourier Transform algorithm.



4.2.17 Epicycles

How the animation in Chapter. 1 works.



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