The Piece-wise Exponential

Shiven Bajpai

AI24BTECH11030
IIT Hyderabad

November 11, 2024

Problem Statement

Given that

$$f(t) = e^{-at}u(t) + e^{bt}u(-t) \tag{1.1}$$

$$u(t) = \begin{cases} 0, & t < 0 \\ \frac{1}{2}, & t = 0 \\ 1, & t > 0 \end{cases} \tag{1.2}$$

$$\int_{-\infty}^{\infty} f(t) = 1 \tag{1.3}$$

Find the possible values of (a, b) if these are the end points of the latus recta of the associated conic. Plot $f(t)$ for these values of (a, b)

Finding the Conic

We expand the integral as

$$\int_{-\infty}^{\infty} f(t) = \int_{-\infty}^{0} f(t) + \int_{0}^{\infty} f(t) \tag{2.1}$$

$$= \int_{-\infty}^{0} e^{bt} + \int_{0}^{\infty} e^{-at} \tag{2.2}$$

$$= \frac{1}{b} + \frac{1}{a} \tag{2.3}$$

Subsituting (1.3) in (2.3)

$$\frac{1}{a} + \frac{1}{b} = 1 \tag{2.4}$$

$$ab - a - b = 0 \tag{2.5}$$

Which is the equation of a conic.

Conversion to Matrix equation

If we take this conic to be of the form, and let $(a, b) = \mathbf{x}$

$$g(\mathbf{x}) = \mathbf{x}^{\mathsf{T}}\mathbf{V}\mathbf{x} + 2\mathbf{u}^{\mathsf{T}}\mathbf{x} + f \tag{2.6}$$

By comparison we obtain:

$$\mathbf{V} = \begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix} \quad \mathbf{u} = \begin{pmatrix} -\frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} \quad f = 0$$

Clearly $\mathbf{V}$ is not of the form of a standard conic. So we will convert it using an affine transformation and then a reflection.

Conversion to a Standard Conic

Let $\mathbf{V} = \mathbf{P}\mathbf{D}\mathbf{P}^{\top}$ (eigen-decomposition). First we apply the affine transformation

$$\mathbf{x} = \mathbf{P}\mathbf{y} + \mathbf{c} \tag{2.7}$$

After this, we get the second eigenvalue as negative, so we use the orthogonal matrix for another transformation

$$\mathbf{y} = \mathbf{P}_0\mathbf{z} \tag{2.8}$$

Here $\mathbf{P}_0$ is the reflection matrix $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

Enpoints of Latus Recta

Finally this gives us a standard conic as

$$\mathbf{z}^{\top}\left(\frac{\mathbf{D_0}}{f_0}\right)\mathbf{z} = 1 \tag{2.9}$$

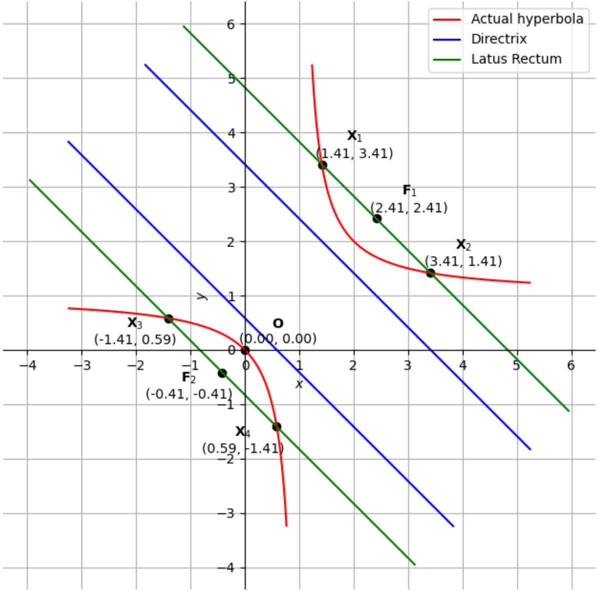
Using standard formulae, we get the endpoints of latus rectum of standard conic ($\hat{\mathbf{z}}$) to be

$$\hat{\mathbf{z}} = \begin{pmatrix} \pm\sqrt{2} \\ \pm 2 \end{pmatrix} \tag{2.10}$$

By reversing the transformations (2.7) and (2.8) we get the endpoints of the actual conic ($\hat{\mathbf{x}}$)

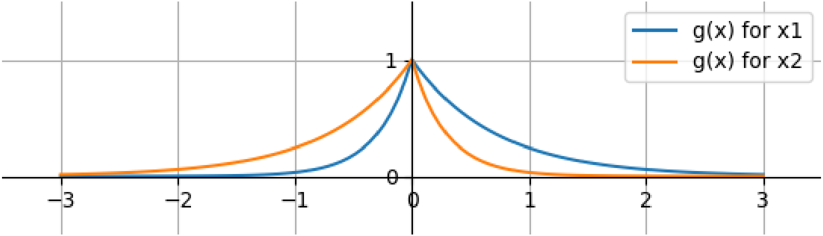
$$\hat{\mathbf{x}}_1 = \begin{pmatrix} 2+\sqrt{2} \\ \sqrt{2} \end{pmatrix} \hat{\mathbf{x}}_2 = \begin{pmatrix} \sqrt{2} \\ 2+\sqrt{2} \end{pmatrix} \hat{\mathbf{x}}_3 = \begin{pmatrix} 2-\sqrt{2} \\ -\sqrt{2} \end{pmatrix} \hat{\mathbf{x}}_4 = \begin{pmatrix} -\sqrt{2} \\ 2-\sqrt{2} \end{pmatrix}$$

Plotting the Conic



Plotting the Function

Out of these, $\hat{\mathbf{x}}_3$ and $\hat{\mathbf{x}}_4$ are unfit to be used as values for (a, b) as the integral in (1.3) will not converge. Plotting f(t) for the remaining two values of (a, b) we get



References

The code for the plot and a fully detailed solution can be found at

https://github.com/shivenBajpai/EE1030/blob/main/Misc/Documentation/main.pdf
https://github.com/shivenBajpai/EE1030/blob/main/Misc/Documentation/Codes/solution.py

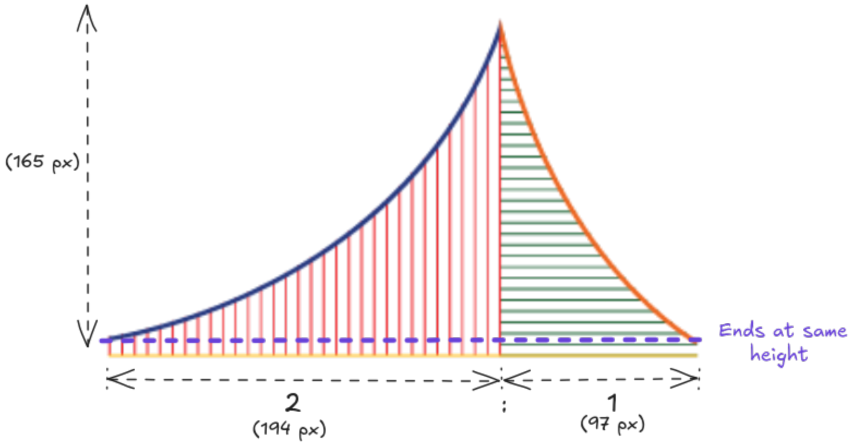
Wait, What logo?

You may have noticed that the function in the last figure somewhat resembles sir's profile picture. That is because it is the same function.

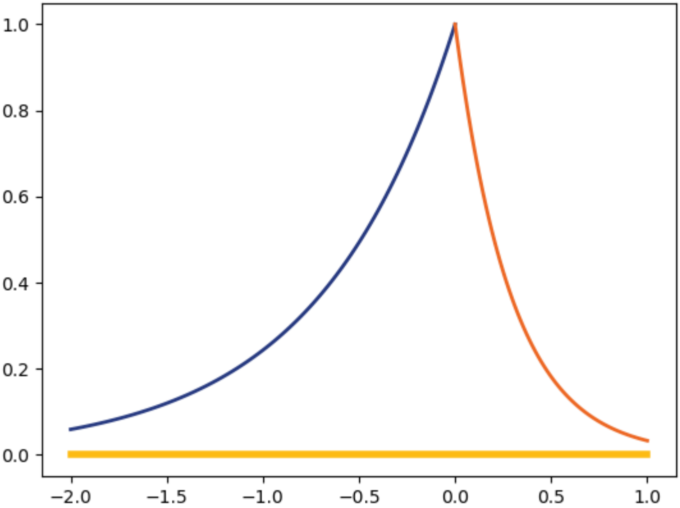
Now we shall seek to recreate that logo.

Observing the original

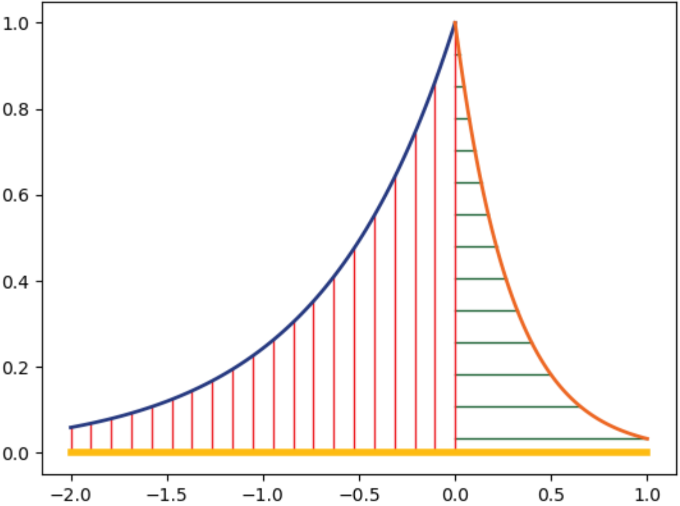
Before we can recreate sir's logo we must first extract some information from the original. Not only do we get the color information, we also find that:



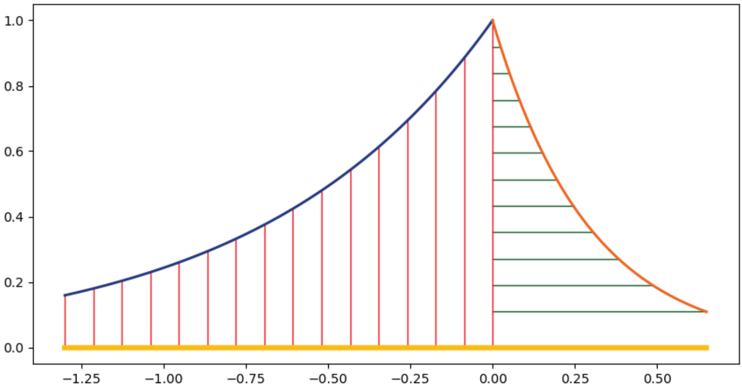
Recreation - Step I: Plot the curve



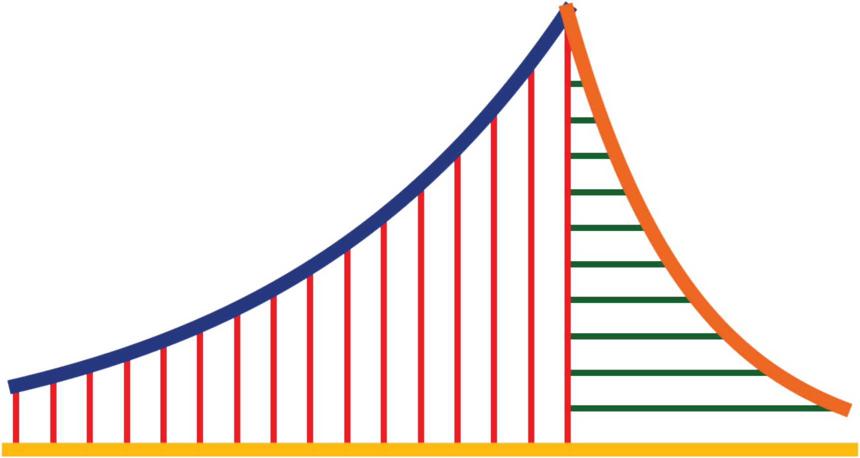
Recreation - Step II: Add the lines



Recreation - Step III: Scaling and Truncation



Recreation - Step IV: Clean it up



A Note on File Formats

Images can be saved in multiple formats. For example .bmp, .jpg/.jpeg, .png, .svg

A Note on File Formats

Images can be saved in multiple formats. For example .bmp, .jpg/.jpeg, .png, .svg

	.bmp	.jpeg	.png	.svg
Type	Raster	Raster	Raster	Vector
Compression	None	Lossy	Lossless	None*
Alpha Channel	No	No	Yes	N/A
Resolution	Fixed	Fixed	Fixed	Infinite

A Note on File Formats

Images can be saved in multiple formats. For example .bmp, .jpg/.jpeg, .png, .svg

	.bmp	.jpeg	.png	.svg
Type	Raster	Raster	Raster	Vector
Compression	None	Lossy	Lossless	None*
Alpha Channel	No	No	Yes	N/A
Resolution	Fixed	Fixed	Fixed	Infinite

Matplotlib can output to all these formats, but you have to specify the transparent parameter of the savefig function as True