Designing a Novel Algorithm for Drawing a Kappa Curve Using Bresenham's Approach

Abdulbasid Banga[1], Nadeem Iqbal[2], Khurram Ejaz[3],

[1]College of Computing and Informatics, Saudi Electronic University, Saudi Arabia
a.banga@seu.edu.sa

[2]Dept. of Computer Science & IT, The University of Lahore, Pakistan
nadeem.iqbal@cs.uol.edu.pk

[3]Dept. of Computer Science & IT, The University of Lahore, Pakistan
khurram.ejaz@cs.uol.edu.pk

Abstract: Bresenham's approach is a classical approach for developing different algorithms. It has already been applied to draw curves like lines, circles, ellipses, parabolas, and hyperbolas. Traditionally, the Kappa curve- Cartesian, parametric, and polar equations draw a mathematical curve. All these three approaches are plagued with some inherent problems. For instance, we can't isolate y from its Cartesian equation if we want to draw it on the display by using the Cartesian equation. Further, both the parametric and polar equations of the Kappa curve contain trigonometric functions which are time-consuming and, of course, run counter to the spirit of interactivity. The figure drawn through these approaches contains inter-pixel spacing. Our proposed algorithm using Bresenham's approach will avoid this spacing and hence the quality of the curve will improve.

Keywords: Kappa curve, Bresenham, pixel, algorithm.

تصميم خوارزمية جديدة لرسم منحنى كابا باستخدام طريقة بريسنهام

الملخص: يعد نهج بريسنهام بمثابة نهج كلاسيكي لتطوير خوارزميات مختلفة. لقد تم تطبيقه بالفعل لرسم منحنيات مثل الخطوط والدوائر والقطع الناقص والقطع المكافئ والقطع الزائد. تقليديا، منحنى كابا – المعادلات الديكارتية والبارامترية والقطبية ترسم منحنى رياضيا. تعاني كل هذه الأساليب الثلاثة من بعض المشاكل المتأصلة. على سبيل المثال، لا يمكننا عزل y عن معادلتها الديكارتية إذا أردنا رسمها على الشاشة باستخدام المعادلة الديكارتية. علاوة على ذلك، تحتوي كل من المعادلات البارامترية والقطبية لمنحنى كابا على دوال مثلثية تستغرق وقتًا طويلاً، وبالطبع تتعارض مع روح التفاعل. يحتوي الشكل المرسوم من خلال هذه الأساليب على تباعد بين وحدات البكسل. سوف تتجنب الخوارزمية المقترحة لدينا باستخدام نهج بريسنهام هذا التباعد وبالتالي ستتحسن جودة المنحنى.

1. Introduction

Computer graphics lies at the intersection of computing and mathematics. Drawing different mathematical curves is a very important field of computer graphics. A lot more approaches exist to draw different curves with their pros and cons. Bresenham's approach is one of the best classical approaches to drawing different mathematical curves (Yi, Zeng et al. 2021). The main strength point of this approach is that it uses only integer arithmetic as far as calculations are concerned (Zhang, Zhang et al. 2022). It avoids the heavy calculations of finding the square roots, cube roots, trigonometric, and other transcendental functions. Of course, this avoidance is of great importance in the sense that the processing speed gets turbo-charged.

A host of mathematical curves exist, some important and some trivial. The Kappa curve is one of them (Popescu, Calbureanu et al. 2021). This curve is symmetrical about both the x-axis and y-axis. Figure 1 shows this curve. Moreover, below is its Cartesian equation:

$$y^2(x^2 + y^2) = a^2x^2 \text{------------------------------} (1)$$

Normally, an equation is converted into explicit form by isolating the y value from the equation. But the problem inherent in this equation is that one can't separate the y value from this equation. This is the major problem to draw the curve of this equation in the computer graphics system. Our strategy will solve this problem. Other approaches also exist to draw this curve like the parametric and polar equations (Blazquez-Salcedo, Doneva et al. 2020). But they also suffer from other problems as well. First, there is an inter-pixel spacing among the different pixels drawn on the graphics system (Bhatnagar, Upadhyay et al. 2023). Further, both the parametric and polar equations contain trigonometric functions that are computationally very heavy consume a lot of precious time of the machine and run counter to the very spirit of interactivity (Mokry 2016). Our methodology will avoid all such kinds of problems.

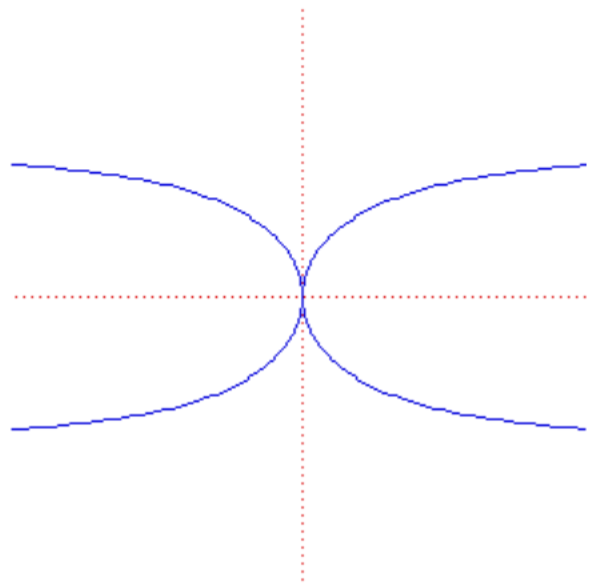

Figure 1. Kappa Curve.

2. Different Approaches for Drawing Kappa Curve

Normally there exist three equations for any curve. These are Cartesian, parametric, and polar equations. Normally, the Cartesian equation is manipulated in the way that y gets placed on the left side and all the other stuff is put on the right side. Once this has been done, the value of independent value is iterated for an arbitrary number of times and the corresponding value for the dependent value is calculated against this equation. This calculated point is drawn on the computer screen. But the problem with this equation is that we can't algebraically isolate the y value from the Cartesian equation. This is the major stumbling block to this equation. Of course, if we are willing to draw its graph on paper, the job will be done by using the symbolic manipulation of the variables but the same can't be done on the computer graphics system.

The second approach is by using the parametric equation. The parametric equations for the Kappa curve are described below:

$$\begin{cases} x = a\cos t \cot t \\ y = a\cos t \end{cases} \text{------------------------------(2)}$$

Below is the algorithm to draw the Kappa curve by using the above equations:

Algorithm 1: *Drawing of Kappa Curve Using Parametric Equations*

for ($\theta \leftarrow 0$ *to* $\theta \le revolution$)

$x \leftarrow a \times \cos\theta \cot\theta$

$y \leftarrow a \times \cos\theta$

drawpixel(x, y)

drawpixel(x, -y)

end for

Figure 2 shows the output of this algorithm.

The third approach is by using the polar equation. In polar equation, there are usually two parameters, *i.e.*, r and θ. These correspondingly are called radius vector and angle.

$$r = a \cot\theta \text{ ------------------------------ } (3)$$

Below is the algorithm to draw the Kappa curve by using the above equations:

Algorithm 2: *Drawing of Kappa Curve Using Polar Equations*

for ($\theta \leftarrow 0$ *to* $\theta \le revolution$)

$r \leftarrow a \times \cot\theta$

$x \leftarrow r \times \cos\theta$

$y \leftarrow r \times \sin\theta$

drawpixel(x, y)

drawpixel(x, -y)

end for

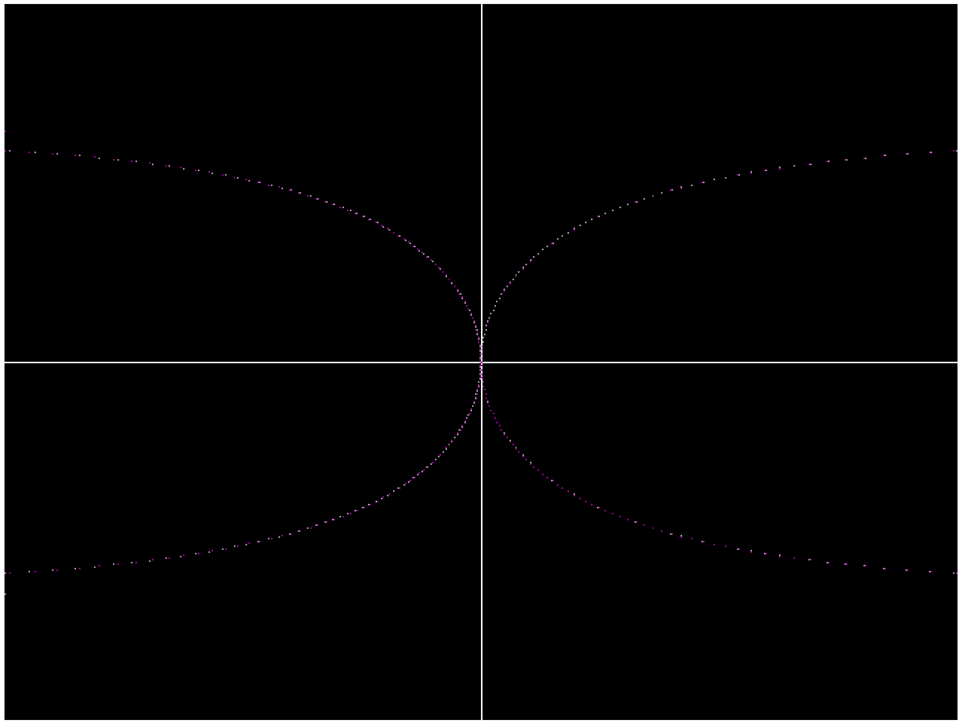

Figure 2. Kappa Curve is generated by using the Parametric Equations.

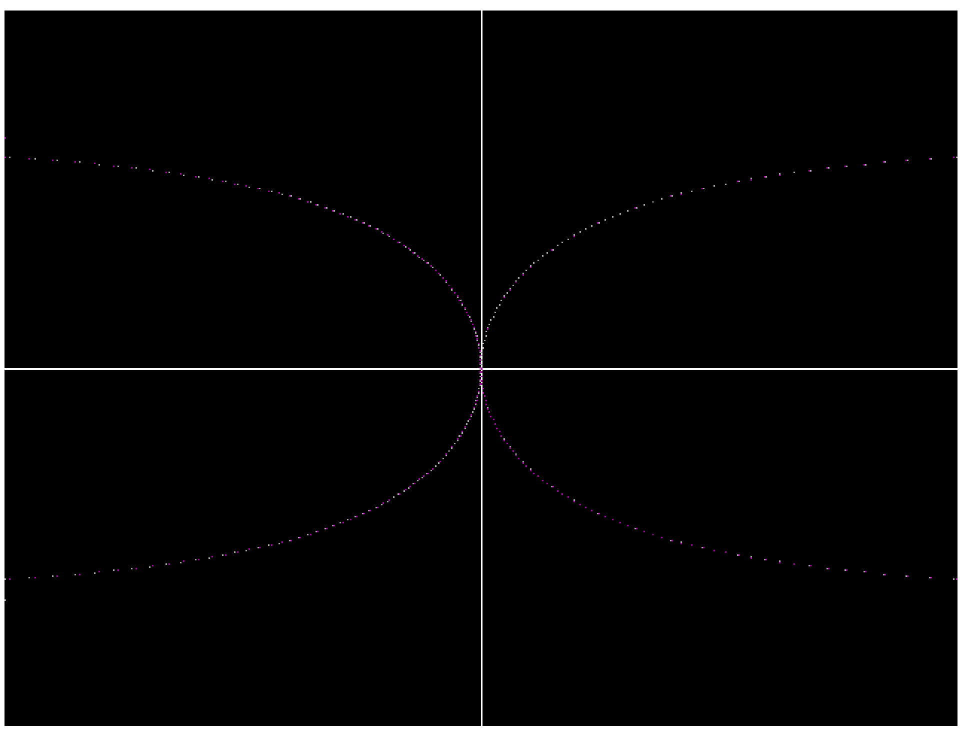

Figure 3. Kappa Curve is generated by using the Polar Equation.

3. Literature Review

In (Cao, Liu et al. 2020), four algorithms have been given namely line, circle, parabola and hyperbola. The later three are also sometimes called conic sections (Nienhaus, Smith et al. 2019). They have also incorporated antialiasing in it. One of the important features of these algorithms is that they have used only integer arithmetic. The problem with the previous approaches was that there was spacing among the pixels. Now with this new methodology, that space problem has been addressed. In yet another area, a 3D warping technique has been discussed at length in (Walia and Verma 2012). The main characteristic of this technique is that it requires small input data. Later, through the experimental results, they demonstrated that their new technique was computationally efficient. Parabola has been scan converted on hexagonal grid as well as discussed in (Prabukumar and Ray 2013). The main salient feature of their research work is that they have used the mid-point approach in their work. In this approach, a pixel closest to the real mathematical curve is chosen. The other benefit includes the visualization of the design ideas through simulation. In (Ray and Ray 2011), two new algorithms of a line have been designed using the parametric equations. The first one uses floating-point arithmetic and the second one uses integer arithmetic which is of course very efficient in the interactive settings. This work has been completed by borrowing the vector generation algorithm and that of Bresenham's. There is one serious drawback of the Bresenham's approach and that is the intersection of a curve with itself. Here that approach fails miserably. To address this problem, a method of deviation for the implicit curves has been devised in (Ray and Ray 2011). Their algorithm translates the implicit equation of analog curve into algorithm. For some curves, like Folium of Desecrate, integer arithmetic is sufficient whereas for other ones, floating point arithmetic is required. A hexagonal drawing grid has also been used for drawing the mid-point ellipse algorithm. For instance, in (Prabukumar and Ray 2012), they have done this task. Further, a comparison has been made between the conventional ellipse drawing algorithms and of their own contribution.

4. Proposed Methodology using the Mid-Point Strategy

Consider the Cartesian equation of the Kappa curve (Chang 2014):

$$y^2(x^2 + y^2) = a^2x^2 \text{------------------------------} (4)$$

We define a Kappa curve function with parameter a as below:

$$f_{kappa}(x, y) = y^4 + x^2y^2 - a^2x^2 \text{------------------} (5)$$

This function has the following properties:

$f_{kappa}(x, y) < 0$ if (x, y) is below the curve

$f_{kappa}(x, y) = 0$ if (x, y) is on the curve

$f_{kappa}(x, y) > 0$ if (x, y) is above the curve

Now obviously as we start from the origin, we will do sampling along y-axis and will determine whether we must increase the value of x or not.

This process will continue until we reach to the threshold where $\frac{dy}{dx} = 1$.

By taking derivative of the Kappa curve, we will equate $\frac{dy}{dx}$ to 1 which will lead us to the terminating condition for the sampling along y-axis. The moment we reach this point, the first region says R_1 will terminate. At that very point, the second region R_2 will start as can be seen in Figure 4.

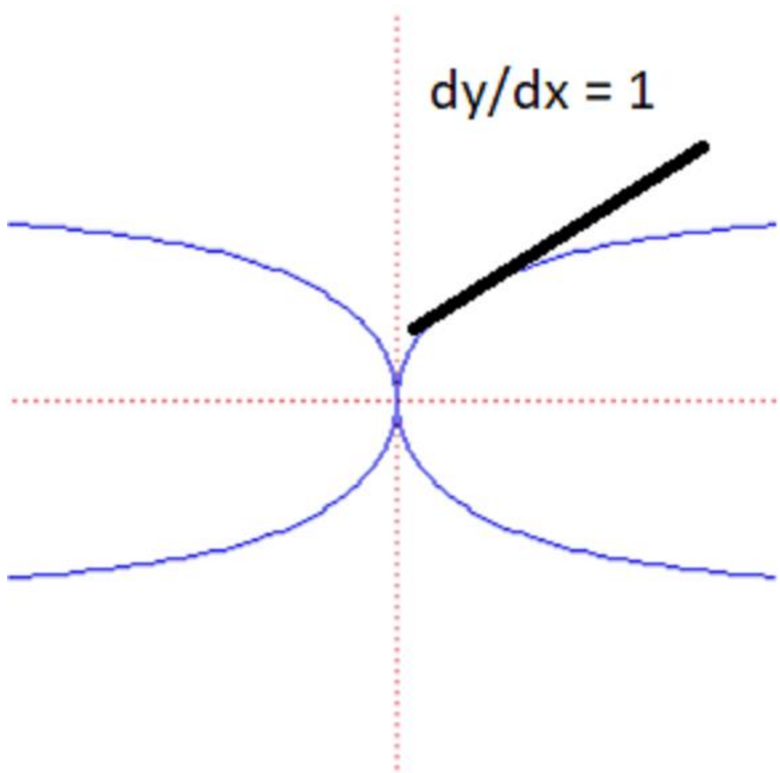

Figure 4. Demarcation of First Region.

The Kappa curve slope is calculated from equation 1 as:

$$\frac{dy}{dx} = \frac{a^2x - xy^2}{x^2y + 2y^3} \text{----------------------------------}(6)$$

At the boundary between R1 and R2,

$$\frac{dy}{dx} = 1 \text{--}(7)$$

By equation these equations, we come up with the following terminating condition for the R_1:

$$x^2y + 2y^3 - a^2x + xy^2 \leq 0 \text{------------------------}(8)$$

We define the decision parameter for the R_1 as below:

$$p1_k = f_{kappa}\left(x_k + \frac{1}{2}, y_k + 1\right) \text{-----------------------}(9)$$

$$= (y_k + 1)^4 + \left(x_k + \frac{1}{2}\right)^2 (y_k + 1)^2 - a^2\left(x_k + \frac{1}{2}\right)^2$$

At the next sampling position ($y_{k+1} + 1 = y_k + 2$), the decision parameter for the R_1 will be evaluated as

$$p1_{k+1} = f_{kappa}\left(x_{k+1} + \frac{1}{2}, y_k + 2\right) \text{---------------------}(10)$$

$$= (y_k + 2)^4 + \left(x_{k+1} + \frac{1}{2}\right)^2 (y_k + 2)^2 - a^2\left(x_{k+1} + \frac{1}{2}\right)^2$$

or

$$p1_{k+1} = p1_k + 4(y_k + 1)^3 + 6(y_k + 1)^2 + 4(y_k + 1) + 1 + \left(x_{k+1} + \frac{1}{2}\right)^2 (y_k + 2)^2 - \left(x_k + \frac{1}{2}\right)^2 (y_k + 1)^2 + a^2\left(\left(x_k + \frac{1}{2}\right)^2 - \left(x_{k+1} + \frac{1}{2}\right)^2\right) \text{------------------}(11)$$

Now if $p1_k < 0$ then $x_{k+1} = x_k$

and

$$p1_{k+1} = p1_k + 4(y_k + 1)^3 + 6(y_k + 1)^2 + 4(y_k + 1) + 1 + \left(x_k + \frac{1}{2}\right)^2 (y_k + 2)^2 - \left(x_k + \frac{1}{2}\right)^2 (y_k + 1)^2 \text{-----------------------------}(12)$$

If $p1_k \geq 0$ then $x_{k+1} = x_k + 1$

and

$$p1_{k+1} = p1_k + 4(y_k + 1)^3 + 6(y_k + 1)^2 + 4(y_k + 1) + 1 + \left(x_k + \frac{3}{2}\right)^2 (y_k + 2)^2 - \left(x_k + \frac{1}{2}\right)^2 (y_k + 1)^2 + a^2(\left(x_k + \frac{1}{2}\right)^2 - (x_k + \frac{3}{2})^2) \text{------------------------}(13)$$

Decision parameter is a very important concept for mid-point algorithm. In region R_1, the value of the initial decision parameter is obtained by evaluating the kappa curve function for

$$(x_0, y_0) = (0, 0) \text{-----------------------}(14)$$

$$p1_0 = \frac{1}{4}(5 - a^2) \text{------------------}(15)$$

Over region R_2, we sample along the x-axis and calculate the corresponding y values. The decision parameter for this region is calculated as:

$$p2_k = f_{kappa}\,(x_k + 1, y_k + \frac{1}{2}) \text{---------------------}(16)$$

$$= (y_k + \frac{1}{2})^4 + (x_k + 1)^2(y_k + \frac{1}{2})^2 - a^2(x_k + 1)^2$$

At the next sampling position ($x_{k+1} + 1 = x_k + 2$), the decision parameter for the second region R_2 will be evaluated as

$$p2_{k+1} = f_{kappa}\,(x_{k+1} + 1, y_{k+1} + \frac{1}{2}) \text{------------------}(17)$$

$$= (y_{k+1} + \frac{1}{2})^4 + (x_{k+1} + 1)^2(y_{k+1} + \frac{1}{2})^2 - a^2(x_{k+1} + 1)^2$$

or

$$p2_{k+1} = p2_k + (y_{k+1} + \frac{1}{2})^4 - (y_k + \frac{1}{2})^4 + (x_{k+1} + 1)^2 \left(y_{k+1} + \frac{1}{2}\right)^2 - (x_k + 1)^2 \left(y_k + \frac{1}{2}\right)^2 + a^2((x_k + 1)^2 - (x_{k+1} + 1)^2) \text{-----------------}(18)$$

If $p2_k \geq 0$ then $y_{k+1} = y_k$

and

$$p2_{k+1} = p2_k + \left(y_k + \frac{1}{2}\right)^2 ((x_{k+1} + 1)^2 - (x_k + 1)^2 + a^2((x_k + 1)^2 - (x_{k+1} + 1)^2)\text{-------}(19)$$

If $p2_k < 0$ then $y_{k+1} = y_k + 1$

and

$$p2_{k+1} = p2_k + (y_k + \frac{3}{2})^4 - \left(y_k + \frac{1}{2}\right)^4 + (x_k + 2)^2 \left(y_k + \frac{3}{2}\right)^2 - (x_k + 1)^2 \left(y_k + \frac{1}{2}\right)^2 + a^2((x_k + 1)^2 - (x_k + 2)^2)\text{----------------}(20)$$

upon entering in the region R_2, the initial value is taken as the last value calculated in the region 1 say (x_0, y_0) and the decision parameter for region R_2 is then calculated as

$$p2_0 = f_{kappa} (x_0 + 1, y_0 + \frac{1}{2})\text{---------------}(21)$$

$$= (y_0 + \frac{1}{2})^4 + (x_0 + 1)^2(y_0 + \frac{1}{2})^2 - a^2(x_0 + 1)^2$$

5. Algorithms

1. Input a and obtain the first point on the kappa curve as

$$(x_0, y_0) = (0, 0)$$

2. Find the initial value of the decision parameter in region 1 as

$$p1_0 = \frac{1}{4}(5 - a^2)$$

3. In region R_1, while the condition $x^2y + 2y^3 - a^2x + xy^2 \leq 0$ remains true do the following

   if $p1_k < 0$, the next point on the kappa curve is $(x_k, y_k + 1)$ and

$$p1_{k+1} = p1_k + 4(y_k + 1)^3 + 6(y_k + 1)^2 + 4(y_k + 1) + 1 + \left(x_k + \frac{1}{2}\right)^2 (y_k + 2)^2 - \left(x_k + \frac{1}{2}\right)^2 (y_k + 1)^2$$

   Otherwise, the next point on the kappa curve is $(x_k + 1, y_k + 1)$ and

$$p1_{k+1} = p1_k + 4(y_k + 1)^3 + 6(y_k + 1)^2 + 4(y_k + 1) + 1 + \left(x_k + \frac{3}{2}\right)^2 (y_k + 2)^2 - \left(x_k + \frac{1}{2}\right)^2 (y_k + 1)^2 + a^2(\left(x_k + \frac{1}{2}\right)^2 - (x_k + \frac{3}{2})^2)$$

4. In region R_2, do the following for an arbitrary number of times

   If $p2_k > 0$, the next point of the kappa curve is $(x_k + 1, y_k)$ and

$$p2_{k+1} = p2_k + \left(y_k + \frac{1}{2}\right)^2 ((x_{k+1} + 1)^2 - (x_k + 1)^2 + a^2((x_k + 1)^2 - (x_{k+1} + 1)^2)$$

   Otherwise, the next point on the kappa curve is $(x_k + 1, y_k + 1)$ and

$$p2_{k+1} = p2_k + (y_k + \frac{3}{2})^4 - \left(y_k + \frac{1}{2}\right)^4 + (x_k + 2)^2 \left(y_k + \frac{3}{2}\right)^2 - (x_k + 1)^2 \left(y_k + \frac{1}{2}\right)^2 + a^2((x_k + 1)^2 - (x_k + 2)^2)$$

5. Find and draw the points in the other three quadrants by using symmetry.

6. Result and Discussion

Figure. 5 shows the Kappa curve drawn by using the proposed algorithm.

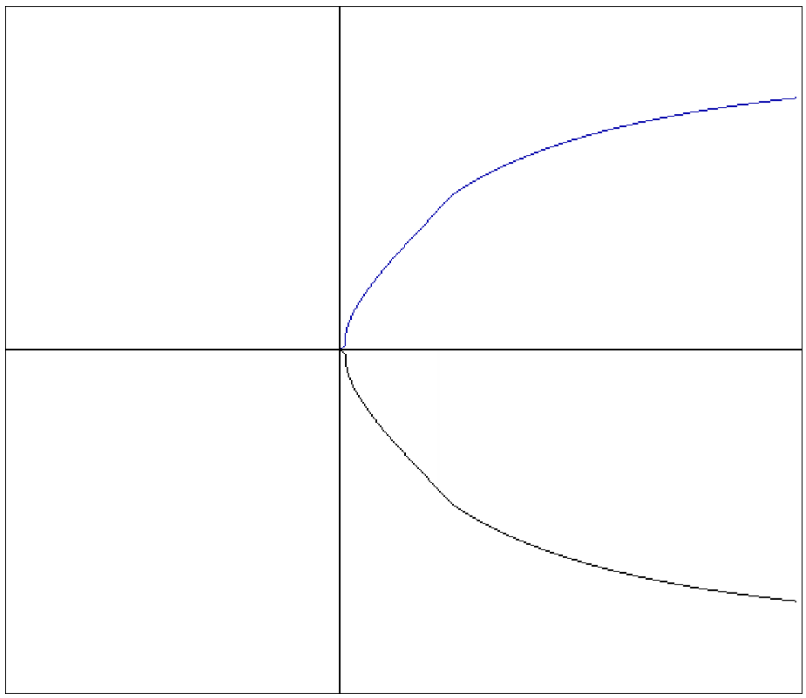

Figure 5. Kappa Curve Drawn through Our Approach.

One can observe a phenomenal difference in the quality of the curves drawn through the traditional approaches of parametric and polar equations and through the one we adopted. There is an inter pixel spacing of the Kappa curve drawn through the traditional approaches. One can clearly see that no such spacing exists in the curve we drawn by using the classical Bresenham approach. Since the Kappa curve is symmetrical about both the x-axis and y-axis, so we just calculated the points in a single quadrant only, i.e., the first quadrant. To draw the pixels for the remaining quadrants, we just exploited this inherent symmetry. Further, we applied calculus to demarcate the two regions. In the first region, there is sampling along y-axis and our algorithm calculates the corresponding x-value. As we reach the threshold, a sampling along x-axis starts and our algorithm calculates the corresponding y-value. Now the questions arises, what will be its terminating condition? Actually, it will never end. Here exists the horizontal asymptote to this curve, so we have put a condition to draw the pixels for some arbitrary number of times.

7. Conclusion and Future Directions

Bresenham is a classical approach to drawing different curves. One of the salient features of this approach is that it uses only integer arithmetic. Further, it always selects the closest pixel along the real mathematical curve to glow a pixel on some calculated position. So, according to this methodology, a curve is drawn by combining different lines. As we have seen, the quality of the same curves drawn by using the parametric and polar equations is not good. A lot more curves exist on which the same methodology can be applied to get different curves with a better quality.

References

[1] Bhatnagar, T., et al. (2023). Pixelated Interactions: Exploring Pixel Art for Graphical Primitives on a Pin Array Tactile Display. Proceedings of the 2023 ACM Designing Interactive Systems Conference.

[2] Blázquez-Salcedo, J. L., et al. (2020). "Polar quasinormal modes of the scalarized Einstein-Gauss-Bonnet black holes." Physical Review D **102**(2): 024086.

[3] Cao, M., et al. (2020). Midpoint distance circle generation algorithm based on midpoint circle algorithm and Bresenham circle algorithm. Journal of Physics: Conference Series, IOP Publishing.

[4] Chang, C.-H. (2014). "Cohen's kappa for capturing discrimination." International Health **6**(2): 125-129.

[5] Mokry, J. (2016). "Recalling prerequisite material in a calculus II course to improve student success." PRIMUS **26**(5): 453-465.

[6] Nienhaus, V., et al. (2019). "Investigations on nozzle geometry in fused filament fabrication." Additive Manufacturing **28**: 711-718.

[7] Popescu, I., et al. (2021). "“Kappa” and “Kieroid” Curves Resulted as Loci." Problems of Locus Solved by Mechanisms Theory: 109-120.

[8] Prabukumar, M. and B. K. Ray (2012). "A mid-point ellipse drawing algorithm on a hexagonal grid." International Journal of Computer Graphics **3**(1): 17-24.

[9] Prabukumar, M. and B. K. Ray (2013). "An Efficient Scan Conversion of Parabola on Hexagonal Grid."

[10] Ray, K. and B. Ray (2011). "An algorithm for Line Drawing Using Parametric Equation." International Journal of Computer Graphics, IJCG **2**(1): 9-16.

[11] Ray, K. S. and B. K. Ray (2011). "A method of deviation for drawing implicit curves." International Journal of Computer Graphics **2**(2): 11-21.

[12] Walia, E. and V. Verma (2012). "A computationally efficient framework for 3D warping technique." International Journal of Computer Graphics **3**(1): 1-10.

[13] Yi, Z., et al. (2021). "A real-time touch control system design based on field-programmable gate array via optimizing Bresenham algorithm for electrowetting displays." Journal of the Society for Information Display **29**(7): 573-583.

[14] Zhang, Y.-x., et al. (2022). "Comparative analysis of DDA algorithm and Bresenham algorithm." International Journal of Intelligent Internet of Things Computing **1**(4): 263-272.