



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE MATEMÁTICA
LABORATORIO DE COMPUTACIÓN GRÁFICA Y GEOMETRÍA APLICADA

Despliegue de Curvas Algebraicas en Coordenadas Baricéntricas.

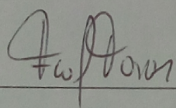
*Trabajo Especial de Grado presentado ante la
ilustre Universidad Central de Venezuela por
el **Br. Luis Rivas** para optar al título de
Licenciado en Matemática.*

Tutor: Dr. Francisco Tovar.

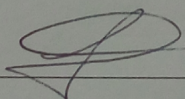
Caracas, Venezuela

Octubre, 2019

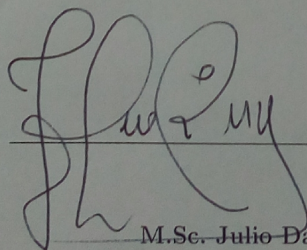
Nosotros, los abajo firmantes, designados por la Universidad Central de Venezuela como integrantes del Jurado Examinador del Trabajo Especial de Grado titulado “**Despliegue de Curvas Algebraicas en Coordenadas Baricéntricas**”, presentado por el **Br. Luis Rivas**, titular de la Cédula de Identidad **V-18.331.259**, certificamos que este trabajo cumple con los requisitos exigidos por nuestra Magna Casa de Estudios para optar al título de **Licenciado en Matemática**.



Dr. Francisco Tovar
Tutor



M.Sc. Jonnathan Otero
Jurado



M.Sc. Julio Daza
Jurado

DEDICATORIA

Quiero dedicarle principalmente este trabajo a mis padres, Luis Rivas y Morita Ponce, a mi novia Doriermis Kelly y a mi abuela Luisa Guánchez, quienes estuvieron conmigo durante toda esta etapa de mi vida, apoyándome incondicionalmente y sin duda siendo mis figuras a seguir.

AGRADECIMIENTOS

Agradezco primeramente a Dios; agradezco a mis padres y a mi novia por el apoyo y el soporte durante esta etapa de crecimiento personal.

Quiero agradecer a Tony Briceño, Keisber González, Daniel Escobar, Yola Fakhry, Emma Mora y a todas las personas que se cruzaron durante mi carrera profesional, ya que cada uno formó parte de este gran aprendizaje.

Agradezco también a mi tutor Dr. Francisco Tovar, por aceptar trabajar conmigo, tenerme paciencia y darme todo su apoyo durante esta etapa final de mi licenciatura.

RESUMEN

Despliegue de Curvas Algebraicas en Coordenadas Baricentricas.

Br. Luis Rivas.

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Matemática

Basados en el trabajo realizado por K. C. Hui y Z. H. Jiang [1], aplicamos parte de ese trabajo para el despliegue de curvas algebraicas en coordenadas baricéntricas. Se desarrolló un algoritmo que permite visualizar el segmento de curva contenida en el interior de un triángulo que define las coordenadas baricentricas del plano, donde la curva, o partes de ella, yacen dentro del triángulo. Dado el triángulo en el plano, se procede a refinar en cientos de triángulos más pequeños que servirán para evaluar y clasificar según la intersección que tengan con la curva. Su clasificación está dividida en tres partes, Triángulos Semillas, Triángulos Frutas y Triángulos Nulos, los cuales dependiendo de dicha clasificación, tienen procesos de evaluación diferentes ya que existen diversas intersecciones entre la curva y los subtriángulos. Esta clasificación depende en muchos casos de la geometría de la curva, esto es, su curvatura o singularidad. Los subtriángulos finales clasificados como Semilla, son los elegidos para el despliegue de la curva, se creará un segmento de recta entre los puntos de corte que existirán en cada subtriángulo. Los puntos de corte serán aproximados mediante un proceso simple de bisección.

Palabras claves: *Coordenadas Baricéntricas, Método de Bisección, Curvas Algebraicas.*

Índice general

Introducción	1
Capítulo 1. Marco Teórico	5
1. Plano Proyectivo y Coordenadas Baricéntricas.	5
2. Curva Algebraica.	8
3. Puntos Simples y Puntos Singulares.	9
4. Curvatura de una curva.	10
5. Método de Bisección.	11
Capítulo 2. Descripción del Algoritmo	13
1. Subdivisión uniforme del triángulo.	14
2. Clasificación de los triángulos.	17
3. Refinamiento para los triángulos Frutas y el criterio de parada.	23
4. Aproximación de la Curva.	25
Capítulo 3. Resultados.	28
Capítulo 4. Algunas Comparaciones con Maple	32
Conclusión	34
Bibliografía	35

Introducción

Con el objetivo de preparar al lector para una mejor comprensión de este trabajo, se presentarán algunos hechos importantes en la historia de la matemática que son base en el desarrollo y evolución de la graficación de curvas haciendo uso de la computadora.

Las coordenadas baricéntricas, introducidas por A. F. Möbius en 1827[2][3], como una respuesta a la pregunta sobre qué masas se deben colocar en los vértices de un triángulo para que un punto dado sea el centro de gravedad de estas masas. Esta herramienta ha sido muy usada en el siglo XIX y comienzos del siglo XX para obtener resultados sobre la geometría del triángulo. Las curvas algebraicas fueron consideradas desde el Renacimiento, período tras el cual, mediante nuevos contenidos y métodos, se han enriquecido teóricamente. Su origen y generación tienen sus causas en el desarrollo de problemas matemáticos históricamente importantes y en numerosas aplicaciones de las mismas en campos como la perspectiva, la óptica, la astronomía, la arquitectura, la cinemática, la mecánica y la tecnología. Tales curvas son representadas de forma implícita y, en algunas ocasiones, también de forma paramétrica. A inicios de la década de los 60 surge una nueva disciplina: *Diseño Geométrico Asistido por Computadoras*, más conocido por sus siglas en inglés **CAGD** (*Computer Aided Geometric Design*), como resultado de la unión de la Geometría y la Computación, y a partir de los trabajos de P. Bézier y P. de Casteljau hubo una revolución en esta área de conocimiento.

Sin embargo, el uso de las curvas algebraicas definidas implícitamente en la solución de problemas de CAGD comenzó hasta mediados de la década de los 80. Dentro de éstas son de especial importancia las llamadas por Chandrajit Bajaj con el nombre de **curvas A-Spline o Splines Algebraicos**, que no son más que curvas definidas por tramos donde cada sección es una curva algebraica definida implícitamente, conectadas con cierto grado de suavidad. Puesto que hay dos tipos de curvas: paramétricas e implícitas, un spline puede construirse con curvas de uno u otro tipo. Los splines que consisten de curvas paramétricas

han sido estudiados a lo largo de los últimos cincuenta años, teniendo mucha importancia las curvas de Bézier. Por otra parte, los splines formados por curvas dadas por ecuaciones implícitas no habían recibido mucha atención por parte de las investigaciones. En 1985 Sederberg[4] inicia el estudio de las A-Splines, los que resultan ser una herramienta muy atractiva en la modelación geométrica, pues no se requieren cálculos vinculados con cambios de parametrización. A partir de estos estudios, comenzaron a aparecer una gran cantidad de trabajos[5],[6] que utilizan tales curvas para el diseño gráfico, la solución numérica de ecuaciones diferenciales, la interpolación y suavizamiento de datos, el ajuste de contornos, entre otros. Todos estos trabajos se basan en las múltiples ventajas que tiene la representación implícita en comparación con la paramétrica. Entre las ventajas principales se destaca que el conjunto de curvas definidas implícitamente es estrictamente mayor que la de las parametrizadas racionalmente, es decir, el conjunto de las curvas paramétricas racionales está contenido estrictamente en el conjunto de las curvas implícitas, (esto ocurre para curvas de grado mayor que tres), como ejemplo se tienen algunas cúbicas no singulares definidas implícitamente las cuales no se pueden parametrizar racionalmente. Para un grado n fijo, las curvas algebraicas definidas implícitamente tienen mayor cantidad de parámetros libres que las racionales paramétricas del mismo grado[7].

En la práctica los parámetros libres adicionales que se obtienen al emplear la representación implícita se pueden utilizar para imponer condiciones extra de interpolación o restricciones geométricas, aproximar una curva complicada con el menor número de parámetros, lograr un mejor ajuste de datos, garantizar un mayor orden de suavidad. Usando la representación implícita es muy fácil resolver el problema de determinar si un punto dado se encuentra o no sobre una curva, y en caso de no estar, determinar de qué lado de la curva se encuentra. Otro problema similar es el de determinar los puntos sobre la curva más cercanos a un punto dado en el plano y calcular la distancia entre ellos. La solución de ambos problemas usando la representación paramétrica es mucho más costosa computacionalmente, requiriendo además de una fórmula de inversión.

Una de las desventajas de las curvas algebraicas implícitas que limitaron su empleo es el hecho de que hasta hace poco (relativamente) no se disponía de algoritmos eficientes

para su graficación. Inicialmente, para graficar curvas algebraicas se propusieron algoritmos que introducían parametrizaciones no racionales[8] y que, en consecuencia, eran costosos a nivel de programación. Más adelante, aparecieron otros métodos que calculaban los puntos sobre una curva algebraica definida implícitamente por la ecuación $f(x, y) = 0$, como las soluciones de tal ecuación polinomial. Estos algoritmos fallaban con frecuencia ante las apariciones de puntos singulares o ramas de la curva que estaban muy cercas entre sí[9].

Aprovechando que un sistema de coordenadas baricéntricas es un caso particular de coordenadas proyectivas, utilizamos las herramientas de la geometría proyectiva para enfocar ciertos problemas, fundamentalmente en la visualización de curvas implícitas mediante un algoritmo.

Para esto, estructuraremos este trabajo de la siguiente manera:

- Un primer capítulo que, son bases preliminares necesarias para la comprensión de este trabajo.
- Un segundo capítulo donde se describe el algoritmo desarrollado con el que se estudian las curvas implícitas. Usando el algoritmo de K. C. Hui y Z. H. Jiang[1] para superficies, pero adaptado para curvas implícitas, se establece un triángulo principal en coordenadas baricéntricas, luego a través del algoritmo se verifica el cumplimiento de ciertas condiciones que va clasificando posibles intersecciones de la curva implícita con el triángulo en coordenadas baricéntricas. Se realiza un refinamiento del triángulo principal, (cuyos vértices encierran la curva o parte de ella), en triángulos mucho más pequeños, para dar así una clasificación de cada triángulo según su intersección con la curva. Clasificamos los triángulos como **Triángulos Frutas**, **Triángulos Semillas** y **Triángulos Nulos**, luego de este proceso, se realizan unas evaluaciones para tener un conjunto final de **Triángulos Semillas**. Para lograr una buena aproximación del punto de intersección entre la curva y un lado del triángulo evaluado, se realiza un proceso de bisección, para aproximar lo más precisamente el punto de intersección entre los lados del triángulo y la curva. Finalmente, ya obtenidos todos los puntos de corte aproximados de todos los Triángulos Semillas, se traza un segmento de recta entre los dos puntos de corte de

cada triángulo evaluado, es evidente que mientras más pequeños sean los triángulos, más preciso será el despliegue final comparado con la curva evaluada.

- Y un tercer capítulo donde se muestran los resultados obtenidos al desplegar curvas implícitas con numerosos puntos simples o de curvaturas altas que por lo general tienden a presentar dificultades numéricas al momento de realizar su despliegue. También se comparan los resultados obtenidos en este trabajo con las gráficas producidas por el programa de graficación de Maple, y se deja en evidencia la mejora visual que genera el algoritmo creado en este trabajo.

Capítulo 1

Marco Teórico

1. Plano Proyectivo y Coordenadas Baricéntricas.

El **plano proyectivo** \mathbb{P}^2 se define como el conjunto de rectas por el origen de \mathbb{R}^3 menos el origen. También se puede considerar \mathbb{P}^2 como un espacio topológico con la topología cociente de \mathbb{R}^3 dada por la relación de equivalencia:

$$(x, y, z) \sim (x', y', z') \iff \frac{x}{x'} = \frac{y}{y'} = \frac{z}{z'}$$

Usaremos la notación: $\mathbb{P}^2 = \{[x, y, z] : (x, y, z) \neq \bar{0}\}$. Donde $[x, y, z]$ denota la recta de \mathbb{R}^3 que pasa por (x, y, z) . La terna $[x, y, z]$ se denomina **coordenada homogénea de \mathbb{P}^2** .

Las coordenadas homogéneas de un punto no son únicas, por ejemplo, las coordenadas $[x, y, z]$ también se representa por $[\lambda x, \lambda y, \lambda z]$ para $\lambda \in \mathbb{R} - \{0\}$. El plano proyectivo \mathbb{P}^2 se puede representar como una completación de \mathbb{R}^2 . Esto es: \mathbb{P}^2 es la unión disjunta de dos conjuntos, uno de ellos es homeomorfo a \mathbb{R}^2 y el otro es una recta. Específicamente:

$$\mathbb{P}^2 = \{[x, y, z] \in \mathbb{R}^3 : z \neq 0\} \cup \{[x, y, z] : z = 0\}$$

Los puntos de coordenadas homogéneas de la forma $[x, y, 0]$ se llaman **puntos en el infinito**.

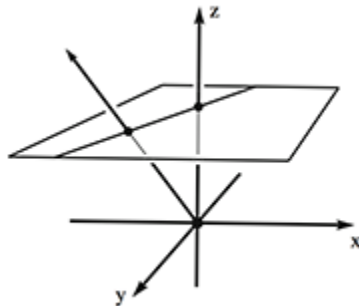


FIGURA 1.1. Representación del Plano Proyectivo

Dadas unas coordenadas homogéneas se pueden construir muchos otros sistemas de coordenadas. Concretamente, para cualquier matriz invertible A ,

$$[x', y', z'] = [x, y, z]A$$

son otras coordenadas homogéneas de \mathbb{P}^2 .

Ahora vamos a producir unas nuevas coordenadas homogéneas especiales.

Consideremos tres puntos no colineales de \mathbb{R}^2 , (x_0, y_0) , (x_1, y_1) y (x_2, y_2) y sean:

$$\begin{cases} a_{11}x + a_{21}y + a_{31} = 0 \\ a_{12}x + a_{22}y + a_{32} = 0 \\ a_{13}x + a_{23}y + a_{33} = 0 \end{cases}$$

Las ecuaciones de las rectas que pasan por (x_1, y_1) y (x_2, y_2) , (x_0, y_0) y (x_2, y_2) y por (x_0, y_0) y (x_1, y_1) respectivamente.

Definimos:

$$s(x, y, z) = \frac{a_{11}x + a_{21}y + a_{31}z}{a_{11}x_0 + a_{21}y_0 + a_{31}}$$

$$t(x, y, z) = \frac{a_{12}x + a_{22}y + a_{32}z}{a_{12}x_1 + a_{22}y_1 + a_{32}}$$

$$u(x, y, z) = \frac{a_{13}x + a_{23}y + a_{33}z}{a_{13}x_2 + a_{23}y_2 + a_{33}}$$

(s, t, u) es un nuevo sistema de coordenadas homogéneas en \mathbb{P}^2 que satisface:

$$s(x_0, y_0, 1) = t(x_1, y_1, 1) = u(x_2, y_2, 1) = 1$$

y $s(x, y, 1) = 0$ es la recta que pasa por (x_1, y_1) y (x_2, y_2) . Análogamente $t(x, y, 1) = 0$ y $u(x, y, 1) = 0$.

Las coordenadas (s, t, u) tienen además la siguiente propiedad. Para cualquier (x, y)

de \mathbb{R}^2 se tiene:

$$(1.1) \quad (x, y, 1) = (s, t, u) \begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{bmatrix}$$

Para mostrar este hecho, note que ambos lados de (1.1) son funciones afines de \mathbb{R}^2 en \mathbb{R}^3 , entonces para verificar que son iguales es suficiente chequear que coinciden en tres puntos no colineales. Tomando $(x, y) = (x_0, y_0)$ obtenemos que debe satisfacerse:

$$(x_0, y_0, 1) = (s(x_0, y_0, 1), t(x_0, y_0, 1), u(x_0, y_0, 1)) \begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{bmatrix}$$

Pero el segundo miembro es igual a:

$$(1, 0, 0) \begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{bmatrix}$$

para terminar la prueba, se evalúa en los puntos $(x_1, y_1, 1)$ y $(x_2, y_2, 1)$.

La ecuación (1.1) muestra que cada uno de los puntos (x, y) de \mathbb{R}^2 se puede presentar unívocamente por medio de una terna (s, t, u) tal que $s + t + u = 1$, en vista de que (1.1) es equivalente a:

$$(x, y) = (x_0, y_0)s + (x_1, y_1)t + (x_2, y_2)u$$

se tiene que (s, t, u) son las **Coordenadas Baricéntricas** de (x, y) con respecto a los puntos (x_0, y_0) , (x_1, y_1) y (x_2, y_2) .

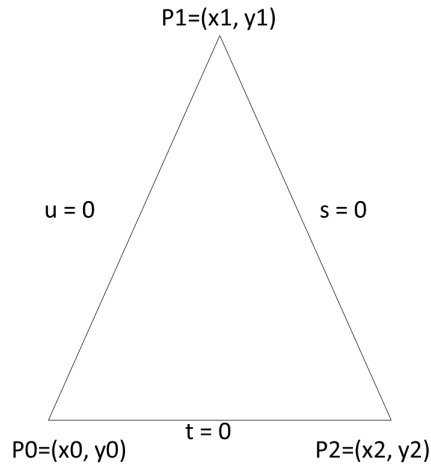


FIGURA 1.2. Coordenadas Baricéntricas

2. Curva Algebraica.

Las curvas algebraicas son representadas de forma implícita. Una **curva algebraica de grado** n definida implícitamente es el conjunto:

$$\{(x, y) \in \mathbb{R}^2 : F(x, y) = 0\}$$

donde

$$F(x, y) = \sum_{0 < i+j \leq n} c_{ij} x^i y^j$$

es un polinomio de grado n con coeficientes reales c_{ij} . De forma más general, sobre un cuerpo \mathbb{K} algebraicamente cerrado.

Consideremos \mathbb{R} y construyamos un espacio afín:

$$\mathbb{R}^n = \mathbb{R}x\mathbb{R}x \cdots x\mathbb{R}$$

que llamaremos **espacio afín n-dimensional sobre \mathbb{R}** . Así \mathbb{R} es la recta afín, el espacio \mathbb{R}^2 es el plano afín y \mathbb{R}^3 es el espacio afín tridimensional.

Si $F \in \mathbb{R}^n[x_1, x_2, x_3, \dots, x_n]$ no es constante, entonces, el conjunto de los ceros de F , $V(F) \in \mathbb{R}^n$ se llama **hipersuperficie algebraica n-dimensional**. Si el grado de dicho polinomio F fuera uno, entonces se diría que F es un **hiperplano n-dimensional**.

Si $\text{Grad}(F) = 1$ y $n = 3$, entonces $F \in \mathbb{R}[x_1, x_2, x_3]$ es un plano del espacio ordinario.

Si F_1, F_2, \dots, F_{n-1} son polinomios de $\mathbb{R}[x_1, x_2, x_3]$ primos entre sí, es decir, donde no tienen ningún factor polinomial en común y sus coeficientes son primos entre sí, entonces el conjunto algebraico $V(F_1, F_2, \dots, F_{n-1})$ se llama **curva algebraica de \mathbb{R}^n** , o bien, **curva algebraica n-dimensional**.

Si $n = 2$ entonces, para un $F \in \mathbb{R}[x_1, x_2]$ el conjunto algebraico $V(F)$ se llama **curva plana algebraica**. Y si el grado de F fuere 1, entonces la curva sería una línea recta del plano, es decir, serían los ceros de un polinomio del tipo $ax_1 + bx_2 + c = 0$.

3. Puntos Simples y Puntos Singulares.

Dada una curva algebraica por el conjunto algebraico $V(F)$, conjunto de los ceros del polinomio F en n indeterminadas, se dice que el punto $P(x_1, x_2, \dots)$ es un **punto simple** si es no nulo en el punto P alguna de las n derivadas parciales con respecto a cada indeterminada:

$$F_{x_i}(P) = \left(\frac{dF}{dx_i} \right)_P \neq 0$$

si se trata de una curva algebraica plana, un punto $P(x_1, x_2)$ será simple si es no nula alguna de las dos derivadas posibles:

$$F_{x_1}(P) \neq 0 \quad \text{o bien} \quad F_{x_2}(P) \neq 0$$

Un punto que no es simple se llama **punto singular** de la curva algebraica. Una curva que solamente está constituida por puntos simples se llama **curva algebraica no singular**.

Tenemos casos particulares con curvas algebraicas que presentan puntos simples o singulares, los cuales representan dificultad al momento de desplegar su gráfico, para evitar este tipo de dificultades se realizan ajustes de modo que el algoritmo pueda desplegar el gráfico de manera eficiente, incluso cuando se trate de curvas singulares.

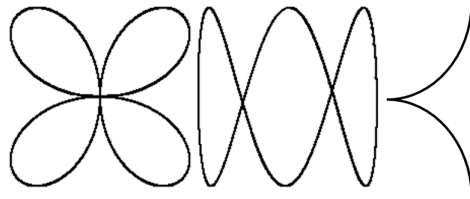


FIGURA 1.3. Ejemplos de curvas con puntos singulares.

4. Curvatura de una curva.

Puesto que una curva definida implícitamente está dada localmente por parametrizaciones, tiene sentido definir su curvatura en un punto. Llamaremos $J : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ a la aplicación lineal que gira un vector 90 grados en sentido antihorario, es decir, $J(x, y)^\top = (-y, x)^\top$. Denotaremos por $H(f)$ a la matriz Hessiana de f definida por:

$$H(f) = \begin{pmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{pmatrix}$$

decimos que la **curvatura de la curva** definida implícitamente por la ecuación $f(x, y) = 0$ y recorrida en el sentido que el gradiente ∇f tenga el mismo sentido que el vector normal n está dada por:

$$k(x, y) = -\frac{J(\nabla f)^\top H(f) J(\nabla f)}{\|f\|^3}$$

Esta definición representa una parte importante en este trabajo, debido a que las curvas con alta curvatura entran en la sección de intersecciones ambiguas, lo que implica una realización de evaluaciones adicionales en el algoritmo desarrollado en este trabajo para un mejor despliegue del gráfico. Esta sección será explicada con mayor cuidado en el capítulo 2.

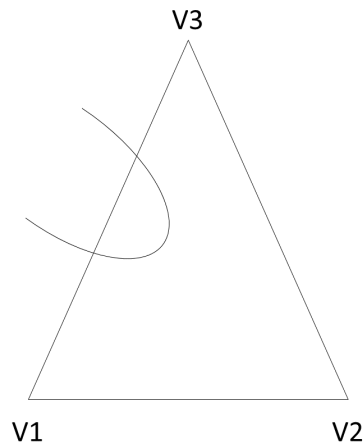


FIGURA 1.4. Ejemplo de una curva con alta curvatura.

5. Método de Bisección.

Como parte importante para el buen desarrollo del algoritmo desarrollado en este T.E.G, se utiliza el método de bisección para aproximar los puntos de intersección de la curva con los lados del triángulo, que definirán los segmentos de recta que servirán para hacer el despliegue de la curva localmente. Conozcamos su funcionamiento.

Esta técnica se basa en el **Teorema del Valor Medio** donde f es continua en $[a, b]$ y parte del supuesto que $f(a)$ y $f(b)$ tienen signos opuestos. Aunque el procedimiento no funciona bien para el caso en el que existe más de una solución en el intervalo (a, b) se considera por simplicidad que es la única raíz de dicho intervalo.

Básicamente, el método consiste en dividir a la mitad repetidamente los subintervalos de $[a, b]$ y en cada paso, localizar la mitad que contiene a la solución m , esto es $f(m) = 0$.

Para empezar, se define $a_1 = a$ y $b_1 = b$ y se calcula el punto medio del intervalo $[a_1, b_1]$ y se denota:

$$m_1 = \frac{a_1 + b_1}{2}$$

Si $f(m_1) = 0$, entonces $m = m_1$, si no, $f(m_1)$ tiene el mismo signo que $f(a_1)$ ó $f(b_1)$. Si $f(m_1)$ y $f(a_1)$ tienen el mismo signo, entonces $m \in [m_1, b_1]$ y se toma $a_2 = m_1$ y $b_2 = b_1$. En el caso contrario, si $f(m_1)$ y $f(b_1)$ tienen el mismo signo, entonces, $m \in [a_1, m_1]$ y se toma $a_2 = a_1$ y $b_2 = m_1$. Luego se repite este proceso al intervalo $[a_2, b_2]$ y así sucesivamente hasta un criterio de parada definido o hasta un mínimo establecido según el margen de error.

Observación: Como en cada iteración, el intervalo es la mitad del intervalo anterior, se puede concluir que en la iteración n , la solución m se encontrará en un intervalo de longitud:

$$Error\ Absoluto = |m - m_n| \leq \frac{b - a}{2^n}$$

para $n \geq 1$. Esto permite tener una idea de que tan cerca se encuentra la solución real, incluso se puede usar para estimar el número de interacciones necesarias para alcanzar una precisión dada.

Nota: En el algoritmo que se desarrolló en este trabajo, se aplica el procedimiento de bisección tres veces como una eurística que da buenos resultados, en la representación del gráfico de $f(x, y) = 0$.

Capítulo 2

Descripción del Algoritmo

Un triángulo T en el plano, es un polígono de tres lados que da origen a tres vértices y a tres ángulos internos. Es la figura algebraica más simple después de la recta. T es denotado por:

$$T = \{V_1, V_2, V_3\}$$

Donde $V_i = (x_i, y_i)$ para $i = 1, 2, 3$, son las coordenadas de los puntos de los tres vértices del triángulo. Cualquier punto P en el interior de T se expresa en coordenadas baricéntricas como:

$$P = [\lambda_1, \lambda_2, \lambda_3]$$

mientras que en coordenadas euclídeas P se expresa de la siguiente forma:

$$P = \lambda_1 V_1 + \lambda_2 V_2 + \lambda_3 V_3$$

donde,

$$\sum_{i=1}^3 \lambda_i = 1 \quad y \quad \lambda_i \geq 0$$

Cualquier punto fuera de T puede también ser expresado en coordenadas baricéntricas, siempre que P sea una combinación afín de los V_i , en este caso, no todos los λ_i son positivos.

El algoritmo que se propone para desarrollar este trabajo, consta de los siguientes pasos:

- (1) Dado un número N , se subdividen los tres lados del triángulo N veces, creando así un refinamiento de N^2 subtriángulos dentro del triángulo principal T , obteniendo de esta manera el primer refinamiento o subdivisión uniforme de T de orden N . Cada subtriángulo se denota por $T_j = [V_{j1}, V_{j2}, V_{j3}]$ para $j = 1, \dots, N^2$.

- (2) Se evalúa la función que define la curva en cada vértice de los triángulos T_j con $j = 1, \dots, N^2$ y se clasifica cada triángulo T_j como **Frutas - Semillas** ó **Frutas - Nulos** (el porqué de estos nombres se especificará más adelante).
- (3) A estos dos conjuntos de subtriángulos, se les realiza una subrutina diferente (los cuales serán explicados más adelante) y de acuerdo a los criterios establecidos en cada subrutina los subconjuntos anteriores se clasificarán en: **Semillas, Frutas y Nulos**. Los triángulos que se clasifican **Nulos** son aquellos que no contienen parte de la gráfica de la función, no se almacenan, simplemente se descartan luego de ser clasificados. Las **Semillas** son los triángulos que contienen, sin ambigüedad, parte del gráfico de la función, y los triángulos clasificados como **Frutas** pasan a un nuevo proceso de refinamiento de orden 2, una vez refinados se realizan los pasos 2 y 3 hasta que desaparezcan las Frutas por completo quedando solo los triángulos clasificados como **Semillas**, o hasta que las Frutas queden con un tamaño considerablemente pequeño de acuerdo a un cierto criterio de parada.
- (4) Finalmente queda un conjunto de subtriángulos **Semillas**. Sobre estos triángulos se realiza un proceso simple de bisección para aproximar el punto de intersección entre la curva y el lado del triángulo y así optimizar el despliegue de la curva.
- (5) Se despliega el gráfico de la curva, el cual consiste de pequeños segmentos de recta contenidos en los subtriángulos que resultaron Semillas.

1. Subdivisión uniforme del triángulo.

Como bien sabemos un triángulo consiste de tres vértices V_1, V_2, V_3 . Una subdivisión de T consiste en particionar $N - veces$ cada lado del mismo, y así, crear N^2 subtriángulos, los cuales quedarán etiquetados y en orden.

Para este proceso, primero se fracciona cada lado del triángulo y se toma cada punto como vértice, tal como se ilustra en la Figura 2.1.

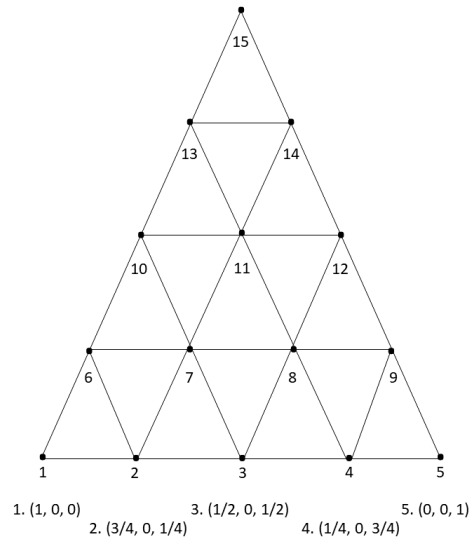


FIGURA 2.1. Discretización del Triángulo para $N = 4$

Teniendo los puntos distribuidos en el triángulo, se construyen los triángulos que "apuntan" hacia arriba, es decir, los triángulos cuya base está en la parte inferior. Asumiendo la misma partición de la Figura 2.1, los triángulos hacia arriba se muestran en el lado izquierdo de la Figura 2.2.

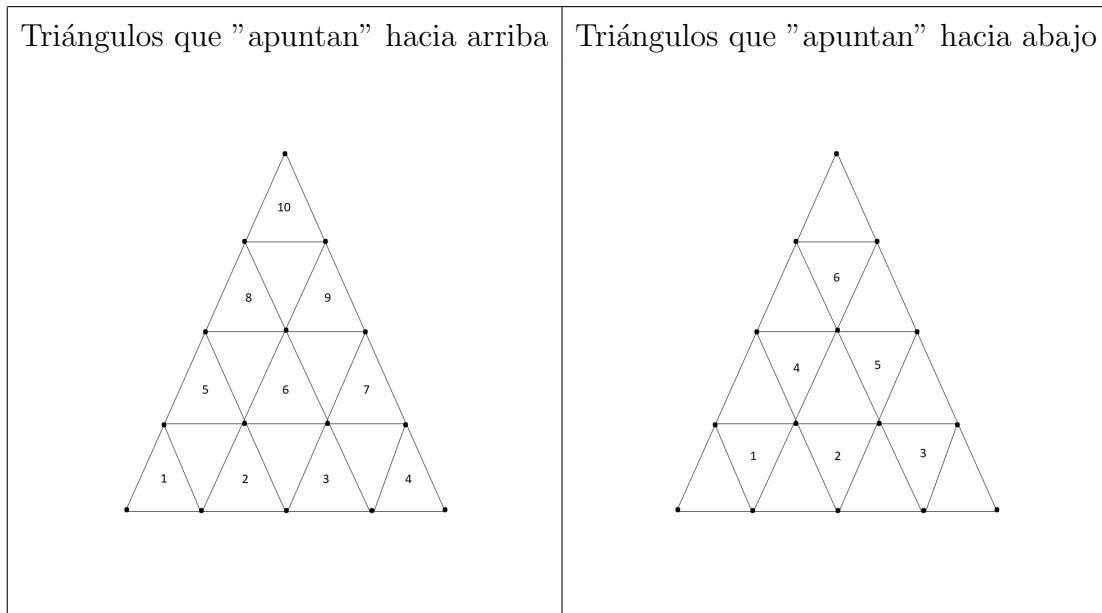


FIGURA 2.2. Construcción de los triángulos hacia arriba y hacia abajo

Se realiza el mismo procedimiento para los triángulos cuya base están en la parte superior (triángulos que "apuntan" hacia abajo) usando la partición anterior en cada lado de los triángulos, tal como se ilustra en el lado derecho de la Figura 2.2.

Una vez identificados los triángulos, se procede con el etiquetado. El etiquetado funciona como guía para evaluar la función en cada triángulo. Su seguimiento será de izquierda a derecha en cada fila del triángulo principal sin importar si el triángulo está hacia arriba o hacia abajo, el etiquetado del primer refinamiento se muestra en la Figura 2.3.

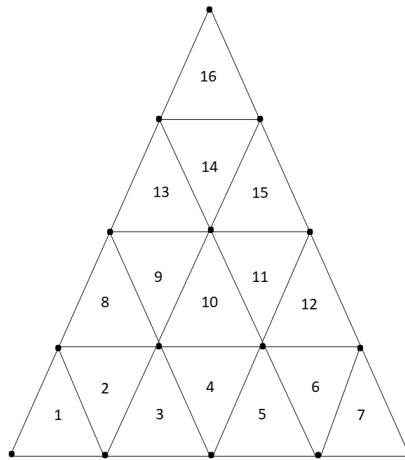


FIGURA 2.3. Etiquetado en orden de los triángulos para $N = 4$

Es importante recalcar, que para realizar este procedimiento, hay que definir el N con el que se va a refinar inicialmente el triángulo principal T , así como también las coordenadas de los vértices del mismo. El etiquetado de todos los subtriángulos que resultan de esta subdivisión, es importante para su posterior uso así como también las coordenadas de cada uno. La unión de este nuevo conjunto de subtriángulos se corresponde con el triángulo más grande, esto significa que:

$$T = \bigcup T_j$$

con $j = 1, \dots, N^2$. Donde T_j denotará el j - *ésimo* triángulo de la subdivisión. En los resultados que se muestran en este trabajo, fue suficiente con definir $N = 50$ ya que el crecimiento del refinamiento de triángulos es exponencial.

2. Clasificación de los triángulos.

Inicialmente, tomando como referencia el trabajo de K. C. Hui y Z. H. Jiang[1], cada subtriángulo T_j , con $j = 1, \dots, N^2$ los clasifican como **nulo**, **fruta** ó **semilla**, a través del siguiente criterio (nótese que al decir que el signo de un vértice de un triángulo es positivo o negativo, nos referimos a la evaluación de la función en dicho vértice):

- a) **Semilla**: Triángulo donde existe un vértice con un signo diferente que el resto y contiene parte de la curva.
- b) **Fruta**: Triángulo que contiene parte de la curva pero, posee una intersección ambigua con la curva (la intersección ambigua será definida más adelante).
- c) **Nulo**: Triángulo cuyos tres vértices son de signos iguales y no hay intersección con la curva.

Debido a las diferentes intersecciones de los triángulo con la curva implícita $F(x, y) = 0$, en este algoritmo se realiza una clasificación inicial de los triángulos T_j en **Frutas - Nulos** y **Frutas - Semillas**, de esta forma resultó un algoritmo diferente al planteado en [1].

2.1. Clasificación Inicial. Como se estableció anteriormente, la subdivisión resultante de T es $\bigcup T_j$ con $j = 1, \dots, N^2$. Los vértices de cada triángulo T_j cumplen una función importante, si un lado $[V_a, V_b]$ de un triángulo cualquiera intersecta a la curva en un único punto, eso quiere decir que la ecuación:

$$\begin{pmatrix} 1 & 1 \\ F(V_a) & F(V_b) \end{pmatrix} \begin{pmatrix} \lambda_a \\ \lambda_b \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

tiene única solución para $\lambda_a, \lambda_b \geq 0$. Un punto de intersección aproximado P existe entre los vértices V_a y V_b sí y sólo si $F(V_a)F(V_b) < 0$ y $P = \lambda_a V_a + \lambda_b V_b$ (esto es según el criterio usado en [1]).

Cabe acotar, que en este trabajo, se optimiza la ubicación de ese punto de intersección P a través de un proceso simple de bisección. A continuación mostraremos cual es la razón para introducir el proceso de bisección.

Ejemplo: La función considerada para este ejemplo es $F(x, y) = x^3 - x^2 + y^2$, y el punto de intersección que vamos a aproximar es el de la función $F(x, y)$ con el eje x en el $(1, 0)$. Ahora bien, al usar el criterio del trabajo de K. C. Hui y Z. H. Jiang[1], tenemos que el punto de intersección está dado por $P = \lambda_a V_a + \lambda_b V_b$, que en este ejemplo resulta ser el punto $P = (1.046, 0)$, ahora usando el proceso de bisección en ese lado del subtriángulo, obtenemos que nuestro punto de intersección es $P = (1.003, 0)$, el cual es una mejor aproximación al punto de intersección real que es: $(1, 0)$. En la Figura 2.4, se puede notar, que hay una mejor aproximación cuando es aplicado el proceso de bisección, es por eso que se realiza este ajuste y es aplicado al final cuando se obtiene un triángulo clasificado como **Semilla**. En el item 2.3 de este capítulo se explica con más detalles esta clasificación. El proceso de bisección es muy simple, ya que solo se usan tres iteraciones por cada punto. La eurística que nos permitió decidir que solo se aplican tres iteraciones, fue porque visualmente se obtienen buenos resultados.

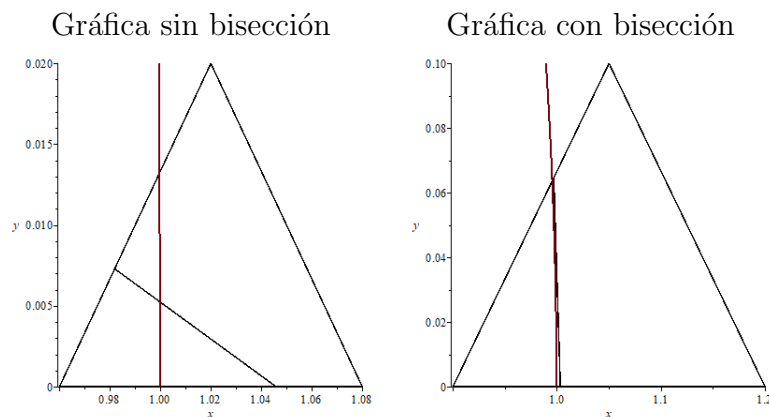


FIGURA 2.4. Diferencia entre no aplicar el proceso de bisección y aplicar el proceso de bisección

Con este ajuste, se tiene que un punto de intersección aproximado P existe entre los vértices V_a y V_b sí y sólo si $F(V_a)F(V_b) < 0$ y P se obtiene con un proceso de bisección. Obviamente, si $F(V_a)$ o $F(V_b)$ es igual a cero, entonces la función $F(x, y) = 0$ pasa por uno de esos vértices. Si la evaluación de ambos vértices es cero, entonces la función yace en ese lado del triángulo o solo está cortando en cada vértice.

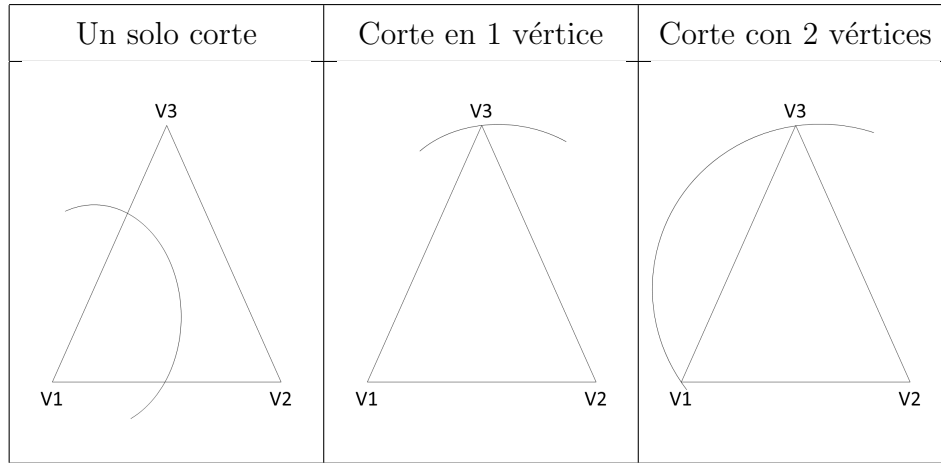


FIGURA 2.5. Intersecciones entre la curva y los triángulos.

Para determinar estas dos primeras preselecciones; se evalúa la función $F(x, y)$ en cada vértice V_{ji} con $j = 1, \dots, N^2, i = 1, 2, 3$, y determinamos que aquellos triángulos donde la evaluación en los tres vértices sean todos mayores iguales que cero (o menores iguales que cero),

$$F(V_{j1}), F(V_{j2}), F(V_{j3}) \geq 0$$

$$(F(V_{j1}), F(V_{j2}), F(V_{j3}) \leq 0)$$

con $j = 1, \dots, N^2$, serán clasificados como triángulos **Frutas - Nulos**, en cualquier otro caso, pasarán a los triángulos **Frutas - Semillas**. En resumen los triángulos **Frutas - Nulos** son aquellos triángulos cuya evaluación de la función en sus vértices tienen el mismo signo.

2.2. Intersecciones ambiguas. Como ya vimos, un punto de intersección único en un lado $[V_{ja}, V_{jb}]$ de un triángulo existe sí y sólo si $F(V_{ja})F(V_{jb}) < 0$. Entonces:

$$\begin{pmatrix} \lambda_a \\ \lambda_b \end{pmatrix} = \frac{1}{F(V_{jb}) - F(V_{ja})} \begin{pmatrix} F(V_{jb}) \\ F(V_{ja}) \end{pmatrix}$$

para $j = 1, \dots, N^2$. Si $F(V_{ja})F(V_{jb}) > 0$, no hay solución con $\lambda_a, \lambda_b \geq 0$. Esto implica que no hay intersección entre el lado $[V_{ja}, V_{jb}]$ y la curva. Sin embargo, esto puede que no sea cierto, como se muestra en la Figura 2.6.

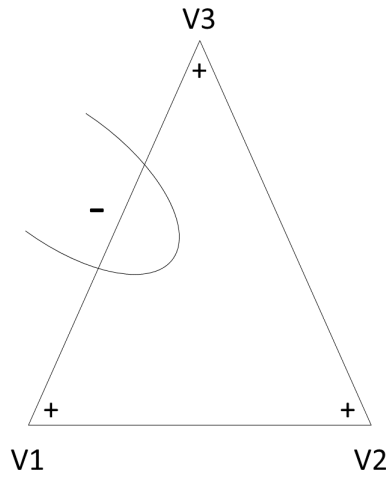


FIGURA 2.6. Ejemplo de una intersección ambigua

Estos triángulos entran en la clasificación de Frutas - Nulos, pues, a pesar de que satisficere la condición $F(V_{ja})F(V_{jb}) > 0$, el triángulo si es intersectado por la curva. Estos casos se corresponden con curvas que tienen una curvatura alta respecto a su entorno y se propone una solución empírica planteada en [1], la cual consiste en evaluar la función en el punto medio de cada lado del triángulo y comparar el signo con los signos de los vértices adjuntos (la evaluación de la función en los vértices), ó comparar el valor absoluto de la evaluación de la función en el punto medio, respecto a los valores absolutos de la función evaluada en los vértices del triángulo.

También existe el caso donde a pesar de entrar en la clasificación de Semillas según

la clasificación dada en [1], el gráfico que arrojaría el algoritmo no se corresponde correctamente con el gráfico de la curva. En la Figura 2.7 se puede apreciar un ejemplo de este caso, donde, a pesar de ser un candidato para semilla, no mostrará un gráfico correcto. Como se puede notar, cuando la curva intersecta al triángulo de esta forma, originalmente se trazaría una línea entre los puntos de corte donde se originan los cambios de signo, de modo que desplegaría una forma incorrecta del gráfico de la curva, como se muestra en el lado derecho de la Figura 2.7.

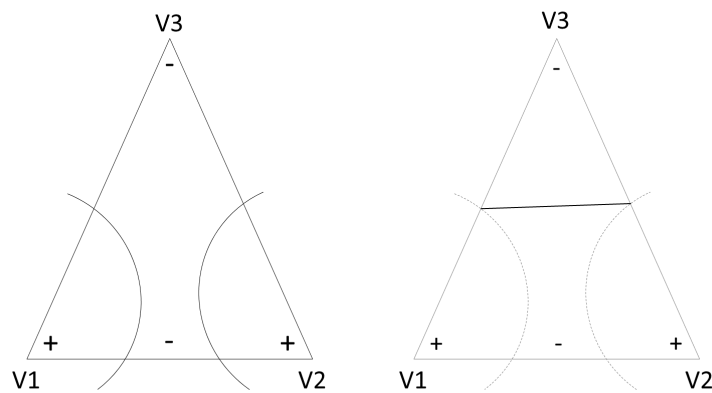


FIGURA 2.7. Despliegue de un segmento de la curva en un triángulo clasificado como Fruta - Semilla sin la clasificación final.

Estos son casos que consideramos como intersecciones ambiguas. Por estos casos se realizan las preselecciones y no se mantiene el patrón original del paper [1].

2.3. Clasificación final. Una vez obtenidos los conjuntos de triángulos Frutas - Nulos y Frutas - Semillas, procedemos a separarlas de forma definitiva a través de dos subrutinas, una para cada conjunto de triángulos.

Para separar los triángulos clasificados como **Frutas - Nulos**, en **Frutas** o (excluyente) **Nulos** se realizan tres evaluaciones:

- a) Se evalúa la función en el punto medio de cada lado del triángulo.

- b) Se comparan los signos de las evaluaciones de los puntos medios con los de la función evaluada en los vértices. Si el signo de la evaluación del punto medio es opuesto al de las evaluaciones de la función en los vértices, entonces ese triángulo pasa a la clasificación **Fruta**.
- c) Los triángulos que no satisfacen el ítem b), se les hace una evaluación adicional. Se comparan los valores absolutos de las evaluaciones de los puntos medios y la evaluación de la función en los vértices, de modo que si cumple lo siguiente:

$$\left| F\left(\frac{V_{ja} + V_{jb}}{2}\right) \right| < \min|F(V_{ja}), F(V_{jb})|$$

$$\left| F\left(\frac{V_{ja} + V_{jb}}{2}\right) \right| > \max|F(V_{ja}), F(V_{jb})|$$

donde $j = 1, \dots, n$ y n es el número total de triángulos clasificados **Frutas - Nulos**. Si los triángulos cumplen con el ítem c) entonces los triángulos también serán clasificados como **Frutas**. Este proceso se realiza con cada combinación de vértices y es usado para distinguir los triángulos que tienen intersecciones ambiguas como en la Figura 2.8 donde, la evaluación del punto medio no satisface el ítem b). Los triángulos **Nulos** simplemente son los que no satisfacen los ítem b) y c) y son descartados en su totalidad. Los triángulos clasificados **Nulos** resultan la mayoría de los triángulos del refinamiento original del triángulo dado.

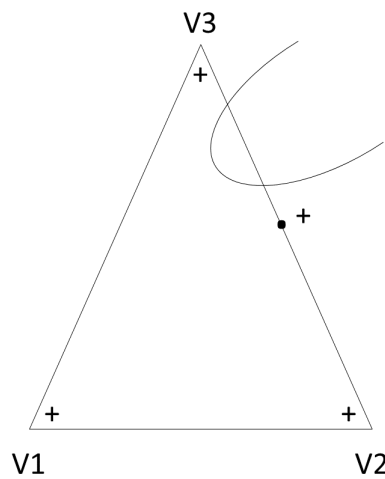


FIGURA 2.8. Intersección ambigua de un triángulo Fruta-Nulo que clasifica como Fruta luego de la evaluación correspondiente al ítem c).

Para separar los triángulos clasificados como **Frutas - Semillas**, en triángulos **Frutas** o (excluyente) **Semillas**, se realizan estos dos pasos de forma simultánea:

- a) Se determina que segmento del triángulo tiene signos iguales en la evaluación de sus vértices.

$$F(V_{ja})F(V_{jb}) > 0$$

para $j = 1, \dots, m$. Donde m es el número total de triángulos clasificados como **Frutas - Semillas**.

- b) Sea $VM_j = F\left(\frac{V_{ja} + V_{jb}}{2}\right)$ la evaluación del valor medio entre los vértices V_{ja} y V_{jb} en el triángulo T_j para $j = 1, \dots, m$, donde m es el número total de triángulos clasificados como **Frutas - Semillas**. Consideramos además, estos dos valores:

- b.1) El valor de la evaluación de la función en el vértice es de signo opuesto al de VM_j .

$$F(V_{ja})VM_j \leq 0$$

- b.2) Ó el valor absoluto de VM_j es menor que el mínimo (ó mayor que el máximo) del valor absoluto de la evaluación de la función en cada vértice de ese lado del triángulo.

$$\begin{aligned} &|VM_j| < \min|F(V_{ja}), F(V_{jb})| \\ &\left(|VM_j| > \max|F(V_{ja}), F(V_{jb})|\right) \end{aligned}$$

para $j = 1, \dots, m$. Si la evaluación del triángulo cumple con los ítems a) y b), (b.1 ó b.2), entonces será clasificado como triángulo **Fruta**. En cualquier otro caso pasa directamente a triángulos **Semillas**, los cuales serán tratados más adelante. Este proceso se realiza con cada combinación de vértices.

3. Refinamiento para los triángulos Frutas y el criterio de parada.

Los triángulos clasificados finalmente como **Frutas** pasan por un refinamiento de orden 2 (o con $N = 2$) para obtener un total de 4 subtriángulos por cada Fruta, y como se especificó en el principio de este capítulo, se toman todos los subtriángulos **Frutas** y se repiten los procesos especificados en el ítem 2 de éste capítulo. Se establece un criterio de parada para este proceso en tres direcciones. La primera es un número finito de refinamiento, la práctica

con el algoritmo ha reportado que un máximo de 15 iteraciones es suficiente. La segunda es el tamaño mínimo de cada triángulo refinado, para determinar el tamaño mínimo, primero definimos el tamaño del triángulo principal, sea T el triángulo principal, sea L_T el máximo de la longitud de los bordes del triángulo principal y sea L_{T_j} el máximo de la longitud de los bordes de un triángulo clasificado como Fruta. Consideremos $\epsilon = \frac{L_T}{10000}$, entonces se establece que cuando se cumple

$$L_{T_j} < \epsilon$$

se detiene el proceso, se considera ese número para ϵ ya que representa un triángulo lo suficientemente pequeño para que no muestre detalles o errores en el despliegue de la curva. La tercera es cuando el conjunto de triángulos Frutas es vacío. Estos criterios son suficientes para obtener un buen refinamiento sobre los triángulos Frutas. Es importante mencionar que, cuando existen curvas con puntos singulares o donde la curvatura es muy alta, quedarán triángulos frutas por refinar, pero no serán un problema ya que el criterio de parada minimiza errores y detalles en el despliegue de la curva. Por esto, es importante determinar el criterio de parada o un máximo de iteraciones para los triángulos clasificados como Frutas.

Observación: El máximo de iteraciones es una medida de seguridad en la programación para evitar que el programa se quede en un lazo infinito o un loop.

En la Figura 2.9 se muestra la diferencia entre la primera subdivisión y la subdivisión de los triángulos Frutas. Debido a que la primera subdivisión tiene una cantidad considerable de subtriángulos, se estableció que la subdivisión uniforme de los triángulos clasificados como Frutas, es de cuatro subtriángulos por cada refinamiento.

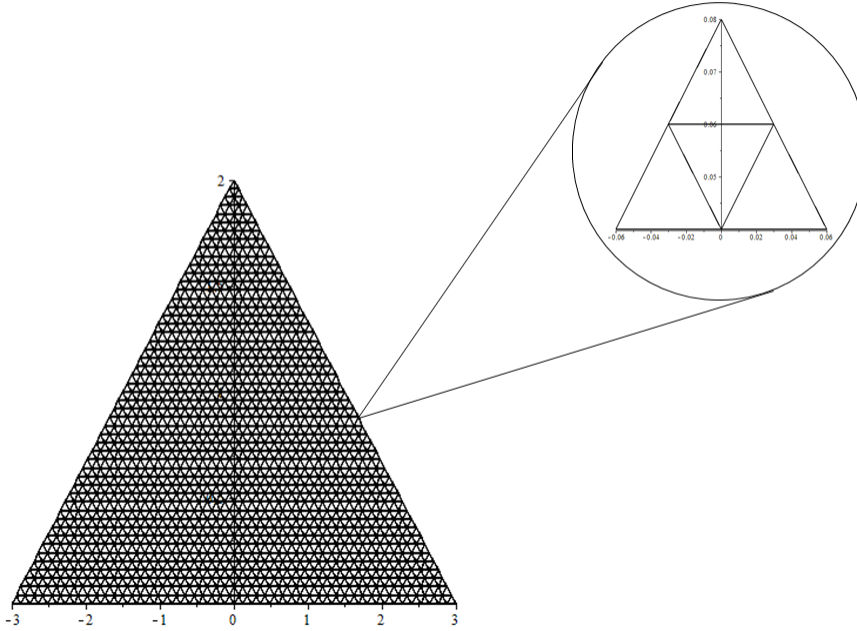


FIGURA 2.9. Se observa el triángulo principal refinado inicialmente con orden 50 o con $N = 50$, dando como resultado 2.500 subtriángulos; mientras que los triángulos clasificados como Frutas cada uno tiene 4 subtriángulos únicamente

4. Aproximación de la Curva.

Como ya vimos anteriormente, un punto P en el interior de T_j , para $j = 1, \dots, N^2$, puede ser expresado como:

$$P = \sum_{i=1}^3 \lambda_i V_i$$

donde,

$$\sum_{i=1}^3 \lambda_i = 1, \lambda_i \geq 0$$

Una función lineal Φ que se aproxima a $F(x, y) = 0$ puede ser expresada como:

$$\Phi(P) = \sum_{i=1}^3 \lambda_i F(V_i), \quad \sum_{i=1}^3 \lambda_i = 1$$

El proceso de bisección mencionado en el capítulo 1, trabaja sobre un intervalo $[a, b]$, como estamos trabajando con triángulos, tenemos tres segmentos posibles en el que realizaremos

dicho proceso. Debido a que todo depende del vértice que tiene el signo diferente y cual es el signo que tiene, se realiza la evaluación de la función en cada vértice del triángulo y se comparan los signos

$$F(V_{ja})F(V_{jb}) < 0$$

para $j = 1, \dots, l$, (l es el número total de triángulos Semillas), para conocer en cual lado se realizará la bisección. La bisección genera un punto de corte más cercano entre la intersección de la curva y el lado del triángulo, eso quiere decir que para cada triángulo Semilla tendremos 2 puntos de corte, entonces al tener dos puntos construiremos el segmento de recta que los une, y se realizará la aproximación lineal de la curva con todos los segmentos de recta que construimos con cada triángulo semilla. El proceso bisección se realiza tres veces en cada segmento del triángulo, puesto que la primera subdivisión del triángulo principal hace un buen refinamiento reduciendo el margen de error de la intersección de la curva con cada lado de los subtriángulos.

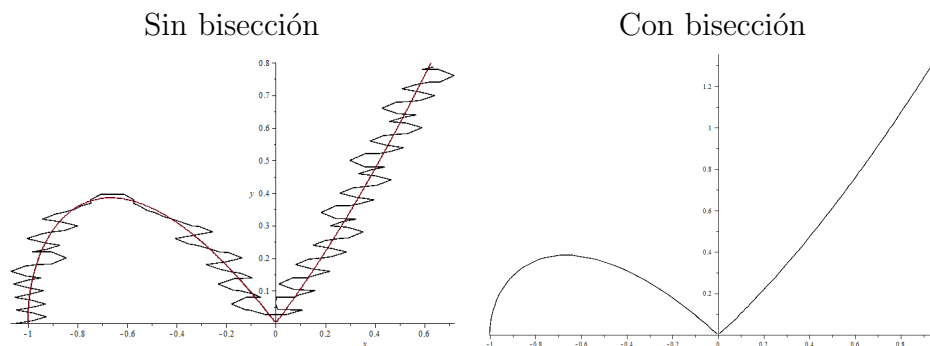


FIGURA 2.10. Diferencia en el despliegue de la función al aplicar el método de bisección.

Como se muestra en la Figura 2.10, el proceso de bisección hace una diferencia considerable en el despliegue del gráfico de la curva. Para la Figura 2.10, se consideró inicialmente $N = 50$ para subdividir uniformemente el triángulo principal (obteniendo un total de 2.500 subtriángulos) y se trabajó con una curva con curvatura alta, que además tiene un punto singular en el $(0, 0)$. La inclusión del proceso de bisección ofrece un aporte importante en el despliegue final. Se nota de manera clara que sólo calculando el punto de intersección por medio la fórmula propuesta por Hui y Jiang [1], no muestra un despliegue correcto de la

curva, de hecho hay una gran diferencia entre el despliegue y la curva original (lado izquierdo de la Figura 2.10). A lo largo del desarrollo del algoritmo, se llega a la conclusión de que 3 procesos de bisección para cada segmento del triángulo donde la curva intersecta al triángulo, es suficiente, ya que el despliegue de la curva se ajusta a la curva original (lado derecho de la Figura 2.10).

En el siguiente capítulo se mostrarán resultados de curvas con curvatura alta y muchos puntos singulares en un solo punto, como también curvas suaves sin puntos singulares, y se comprueba la eficacia del algoritmo en estos tipos de situaciones, realizando con éxito los despliegues de las curvas.

Resultados.

A continuación se muestran resultados obtenidos con el algoritmo propuesto. Se ilustran resultados sobre curvas complejas debido a que tienen puntos singulares y curvatura alta, sobre todo en el punto $(0,0)$ (elegido por comodidad para las visualizaciones de los despliegues). Hay curvas que por su alta curvatura en el punto $(0,0)$ (como en la Figura 3.5) el algoritmo no termina de graficar la curva, sin embargo, en comparación con la rutina de graficación de un programa como Maple para curvas algebraicas, el algoritmo muestra un despliegue que se ajusta mucho mejor a la curva original.

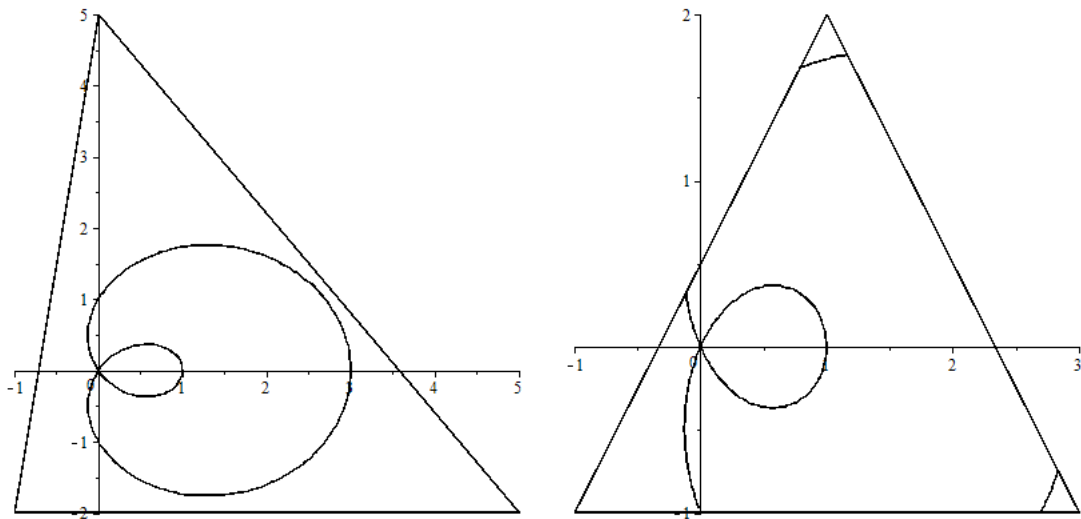


FIGURA 3.1. La curva $(x^2 - 2x + y^2)^2 - (x^2 + y^2) = 0$ (Cardioide), sobre dos triángulos diferentes.

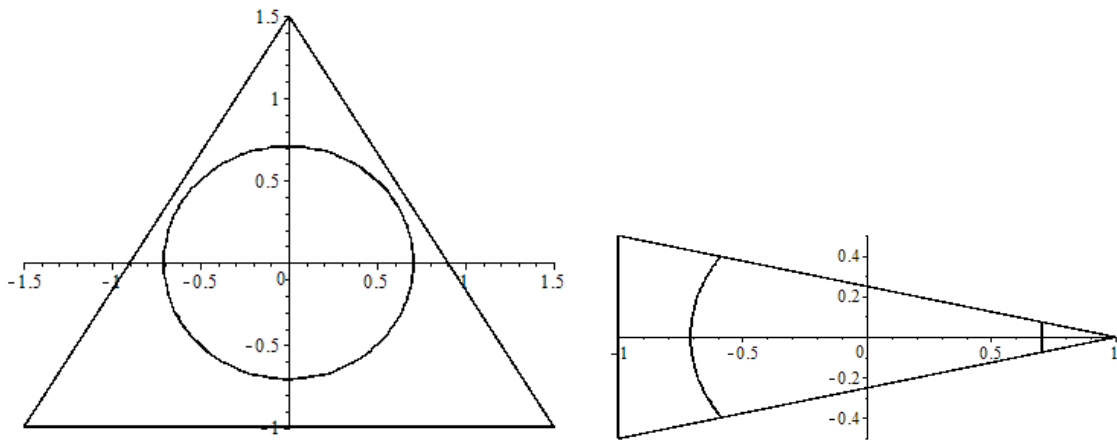


FIGURA 3.2. La curva $x^2 + y^2 - 0,5 = 0$ (Circunferencia), sobre dos triángulos diferentes.

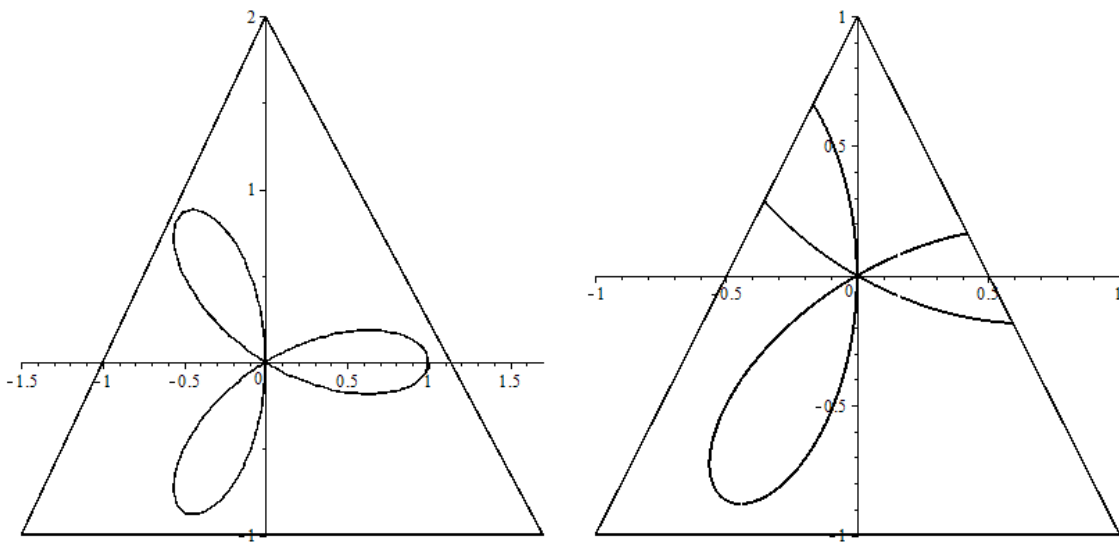


FIGURA 3.3. La curva $(x^2 + y^2)^2 - x(x^2 - 3y^2) = 0$ (Flor de 3 Hojas), sobre dos triángulos diferentes.

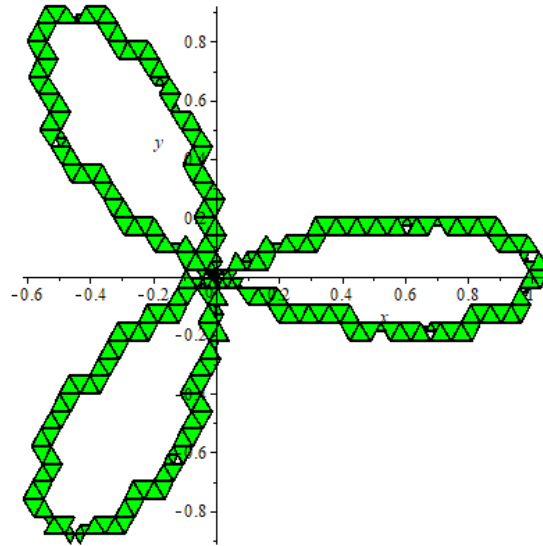


FIGURA 3.4. Conjunto de triángulos clasificados Semillas con refinamiento inicial de orden 50 - Flor de 3 Hojas. Nótese la acumulación de subtriángulos entorno del $(0, 0)$ que es una singularidad de orden 2, como resultado de refinar el despliegue en un entorno de ese punto.

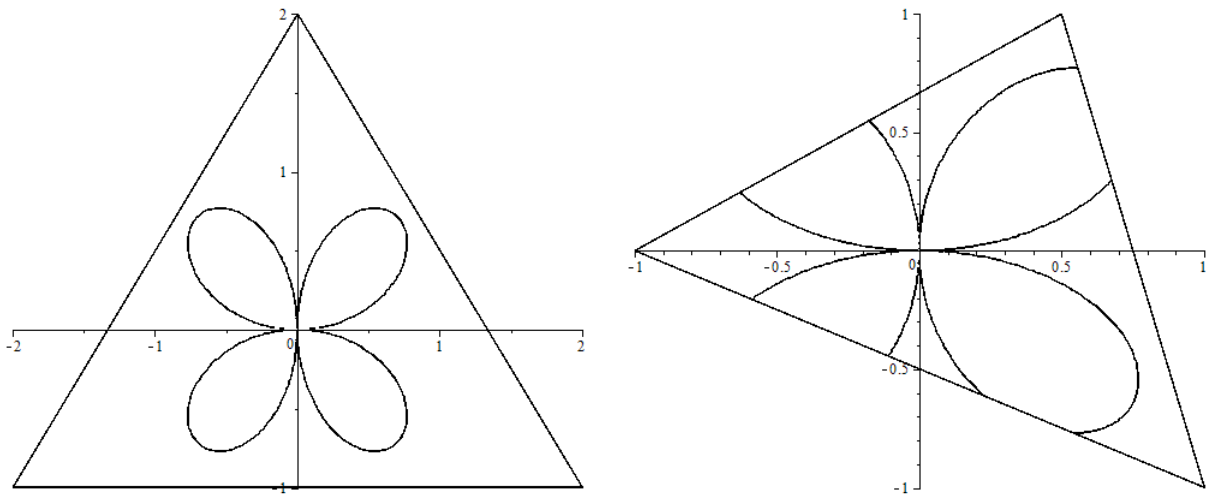


FIGURA 3.5. La curva $(x^2 + y^2)^3 - 4x^2y^2 = 0$ (Flor de 4 Hojas), sobre dos triángulos diferentes.

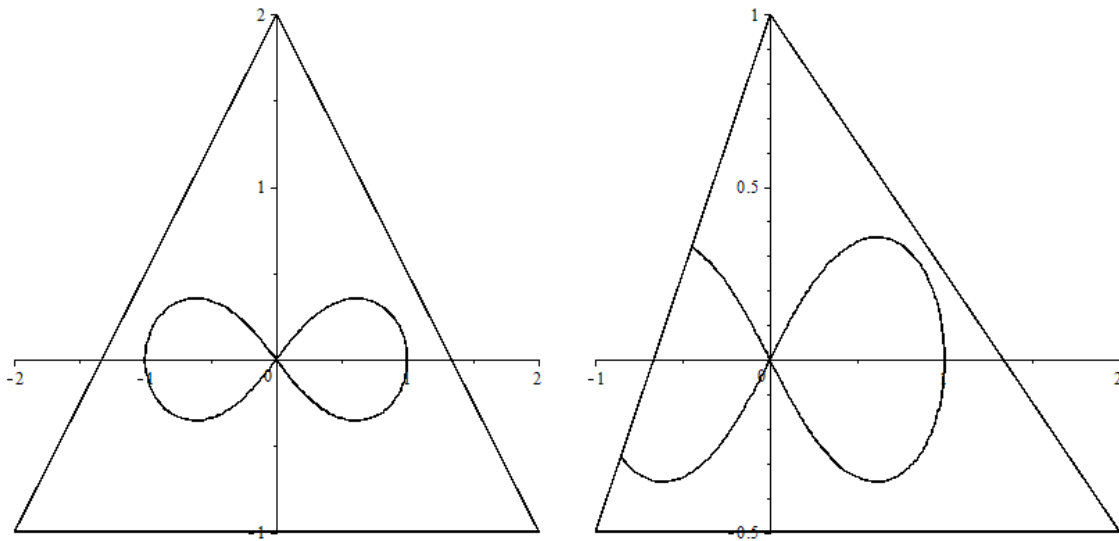


FIGURA 3.6. La curva $(x^2 + y^2)^2 - (x^2 - y^2) = 0$ (Lemniscata), sobre dos triángulos diferentes.

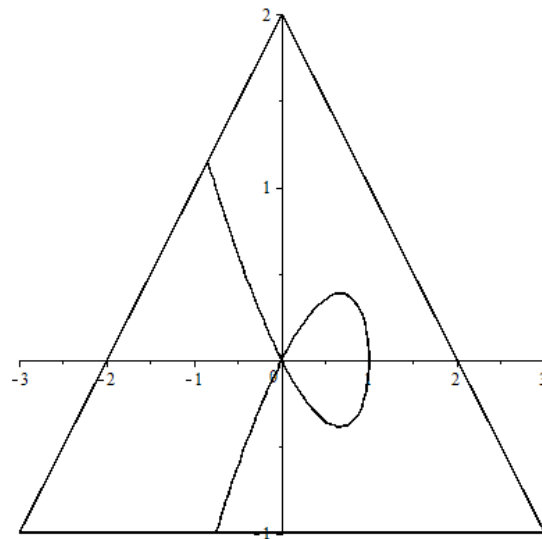


FIGURA 3.7. La curva $x^3 - x^2 + y^2 = 0$, cúbica con singularidad en $(0, 0)$

Capítulo 4

Algunas Comparaciones con Maple

A continuación se comparan los resultados del algoritmo con Maple, en base a los resultados antes expuestos. Cabe destacar, que las curvas en Maple se pueden graficar completas o con un rango específico rectangular, pero no se puede picar la curva en partes, como en efecto si lo puede hacer el algoritmo. Por otro lado, hay curvas que Maple no logra graficar de forma óptima, las cuales el algoritmo las despliega más ajustadas a su forma.

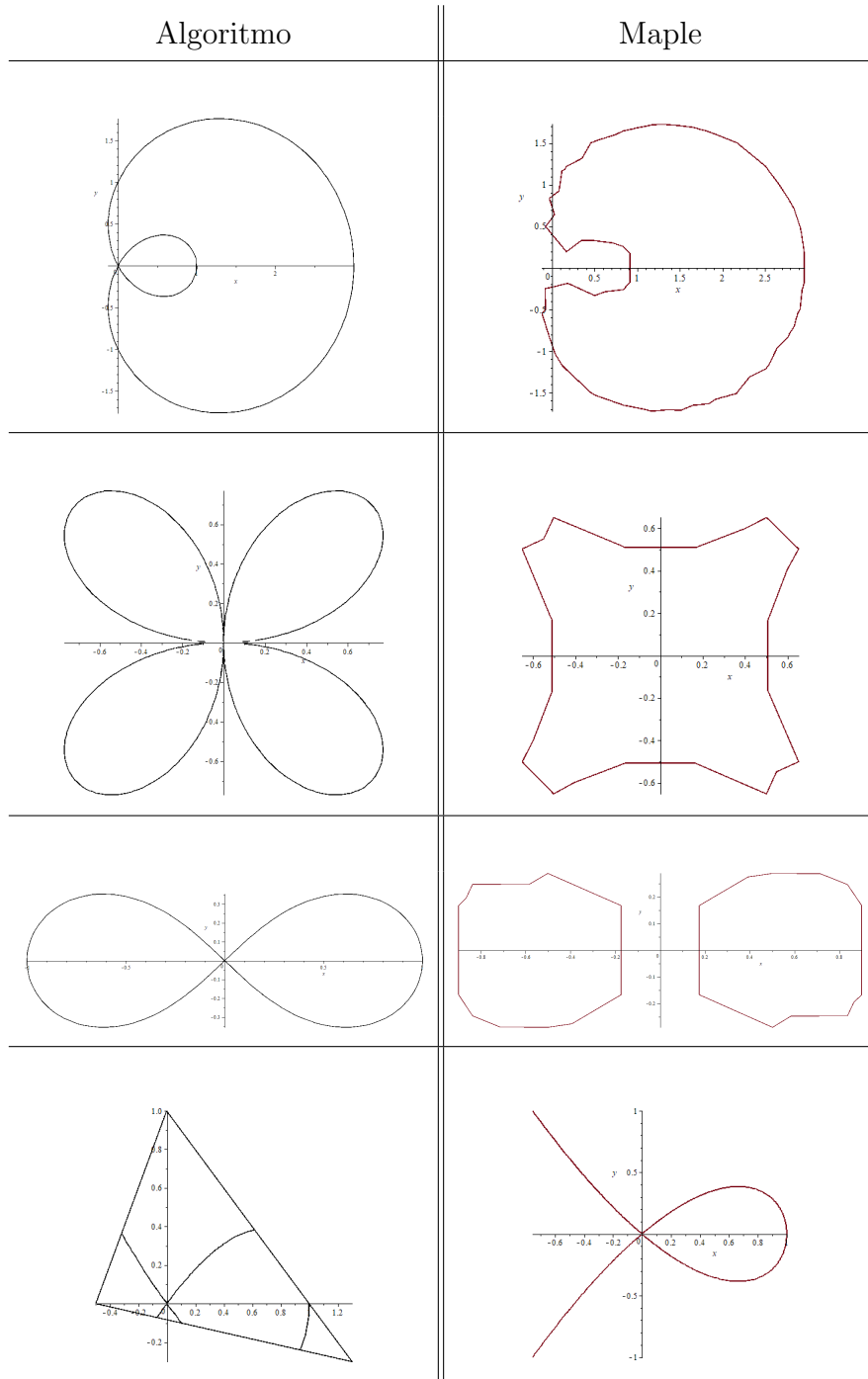


FIGURA 4.1. Comparaciones entre el algoritmo y Maple

Conclusión

El algoritmo de despliegue de curvas algebraicas, el cual evalúa una curva que yace dentro de un triángulo en coordenadas baricéntricas y luego la gráfica mediante segmentos de rectas, es una adaptación del trabajo de K. C. Hui y Z. H. Jiang[1], y fue desarrollado en orden de eliminar discontinuidades, errores de intersecciones ambiguas y aumentando la capacidad de memoria con respecto a cualquier software de graficación computarizada. Mientras el trabajo de [1] no consideran el proceso de bisección, la inclusión de este proceso dió la capacidad de desarrollar un algoritmo de despliegue que opera eficientemente. El algoritmo es particularmente útil para trabajar partes de una curva, intersecciones entre curvas y curvas que poseen curvatura alta, en comparación con el programa de graficación de Maple, el cual queda en evidencia la mejora que realiza el algoritmo al momento de visualizarlas.

Los tiempos de respuesta del algoritmo con respecto a los ejecutores de Mplae no fueron comparados, sin embargo el algoritmo trabaja rápido para curvas complicadas.

Cuando se trabaja con *splines algebraicos* de grado 3, se puede construir un triángulo a partir del cuadrilátero con el que se trabaja para generar el *spline*. Esa es una gran utilidad de trabajar en coordenadas baricéntricas y esto es una aplicación que puede realizar el algoritmo en otro tipo de estudio.

Bibliografía

- [1] K.C. Hui and Z. H. Jiang, Tetrahedra Based Adaptive Polygonization of Implicit Surface Patches. The Chinese University of Hong Kong, vol 18, pp 57-68 (1999)
- [2] Mobius A. F., Der Barycentrische Calcul. Johann Ambrosius, Leipzig, (1827)
- [3] Angel Montesdecoa, Geometría Métrica y Proyectiva en el Plano con Coordenadas Baricéntricas. Algunos Tópicos, Versión 2.181231712
- [4] T. W. Sederberg, Piecewise Algebraic Surface Patches, Computer Aided Geometric Design 2, pp 53-59, (1985)
- [5] C. L. Bajaj, The Emergence of Algebraic Curves and Surface in Geometric Design, Tech. Report Purdue University, CS-TR-92-056, (1992)
- [6] C. L. Bajaj y G. Xu, Data Fitting with Cubic A-splines. Computer Graphics International, CGI94, Melbourne, Australia, (1994)
- [7] G. Farin, Curves and Surfaces for CAGD, Arizona State University, pp 216-218, (2002)
- [8] R. Patterson, Parametrizing and Graphing Nonsingular Cubic Curves, Computer Aided Design 20, pp 615-623, (1988)
- [9] R. E. Chandler, A Tracking Algorithm for Implicitly Defined Curves, IEEE Computer Graphics and Applications, (1988)
- [10] W. Morales, Generación Jerárquica de Puntos, Curvas d-offset, y Ploteo de Curvatura Asociados a un A-Spline Cúbico, Universidad de la Habana, (2012)
- [11] A. Valdés, Notas de Geometría Diferencial con Aplicaciones, pp 47, (2013)
- [12] Documento Digital: <https://tecdigital.tec.ac.cr/revistamatematica/HERRAMInternet/ecuaexecl/node4.html>
- [13] L. Fernández, Plano Afín, Coordenadas Baricéntricas y Plano Proyectivo, E.T.S.I. Navales, Universidad Politécnica de Madrid, E-28040-Madrid.

