

Matemáticas Computacionales

Practica 1: Gráficas de curvas en R

1860560 Rangel Delgado Jesus Angel

16 de Febrero del 2021

1. Introducción

1.1. Desarrollo de la practica

En esta primera práctica se hará una de las cosas básicas al momento de aprender R. Se repasaran las curvas en \mathbb{R}^2 vistas en primer semestre en la materia de Geometría Analítica. [1]. Se graficarán curvas como la recta, parábola, circunferencia, elipse e hipérbola.

2. Curvas de \mathbb{R}^2

2.1. Línea Recta

Def. Llamamos **línea recta** al lugar geométrico de los puntos tales que tomados dos puntos diferentes cualesquiera $P_1(x_1, y_1)$ y $P_2(x_2, y_2)$ de lugar, el valor de m se calcula por medio de:

$$m = \frac{y_1 - y_2}{x_1 - x_2} \quad (1)$$

Ecuación de la recta dada su pendiente y ordenada en el origen
Es la recta cuya pendiente \mathbf{m} y cuya ordenada en el origen es \mathbf{b} , que tiene por ecuación:

$$y = mx + b \quad (2)$$

Código en R

Para poder graficar esta función, solo basta con darle valor a la pendiente \mathbf{m} y su ordenada \mathbf{b} , además falta establecer un límites que serán el extremo al que llegara nuestra función.

Ejemplo 1:

```
1 #Línea recta 1:  
2 m <- 7 #pendiente  
3 b <- 7 #interseccion  
4  
5 #funcion de la línea recta  
6 f <- function(m, b, x){  
7   return(m * x + b)  
8 }  
9  
10 x <- seq(-22, 22, 0.01)  
11 y <- f(m, b, x) #sirve para evaluar  
12  
13 plot(x, y, type = "l", xlab = "Eje x", ylab = "Eje y") #graficamos  
14 abline(h = 3, v = 3)
```

Figura 1: Código de la línea 1

y su gráfica seria:

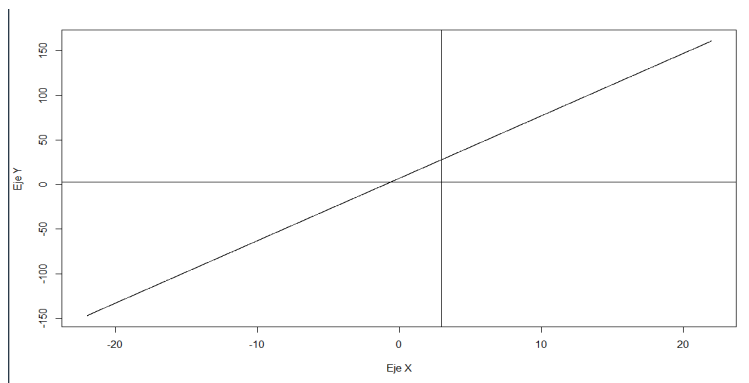


Figura 2: Gráfica del código 1

Ejemplo 2:

```
20 #funcion de la linea recta
21 f <- function(m, b, x){
22   return(m * x + b)
23 }
24
25 x <- seq(-8, 8, 0.01)
26 y <- f(m, b, x)
27
28 plot(x, y, type = "l", xlab = "Eje X", ylab = "Eje Y")
29 abline(h = 1, v = 1)
```

Figura 3: Código de la linea 2

y su gráfica seria:

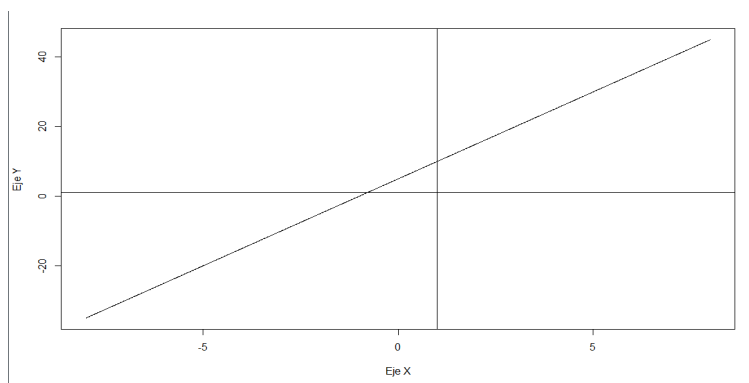


Figura 4: Gráfica del código 2

2.2. Circunferencia

Def. Circunferencia es el lugar geométrico de un punto que se mueve en un plano de tal manera que se conserva siempre a una distancia constante de un punto fijo de ese plano

Teorema La circunferencia cuyo centro es el punto (h, k) y cuyo radio es la constante r , tiene por ecuación:

$$(x - h)^2 + (y - k)^2 = r^2 \quad (3)$$

Código en R Para poder graficar una circunferencia en R primero tenemos que asegurarnos de que nuestra r o sea el radio de la circunferencia sea mayor que cero, después hay que comenzar a graficar por partes, nuestra circunferencia en este caso iniciamos con la parte positiva para después graficar la negativa, por ultimo se asignan valores a nuestra función, los valores para h , k y r .

Ejemplo 1:

```

31 #Circunferencia 1:
32 circunferencia <- function(h, k, r){
33   if (r >= 0){
34     if (r == 0){
35       plot(x = h, y = k, xlab = "Eje X", ylab = "Eje Y")
36     } else{
37       x <- seq(h - r, h + r, 0.01)
38       ypositiva <- k + sqrt(r^2 - ((x - h)^2)) # parte positiva de la circunferencia
39       ynegativa <- k - sqrt(r^2 - ((x - h)^2)) # parte negativa de la circunferencia
40       # graficamos primero la parte positiva
41       plot(x, ypositiva, type = "l", xlim = c(h - (r + 1), h + (r + 1)), ylim = c(k - (r + 1), k + (r + 1)),
42           xlab = "Eje X", ylab = "Eje Y")
43       lines(x, ynegativa, type = "l") # agregamos la parte negativa
44       abline(h = 0, v = 0)
45       points(x = h, y = k, col = "red")
46     }
47   } else{
48     return(print("El radio no es positivo."))
49   }
50 }
51
52 # ejecutamos la función
53 circunferencia(0, 0, 5)

```

Figura 5: Código de la circunferencia 1

y su gráfica seria:

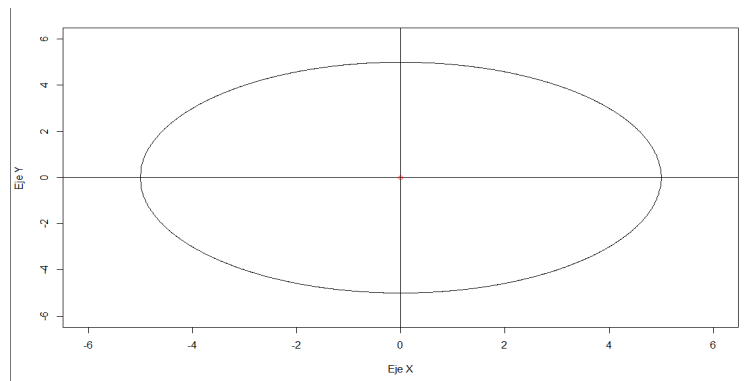


Figura 6: Gráfica de la circunferencia 1

Ejemplo 2:

```

55 #circunferencia 2
56 circunferencia <- function(h, k, r){
57   if (r >= 0){
58     if (r == 0){
59       plot(x = h, y = k, xlab = "Eje X", ylab = "Eje Y")
60     } else{
61       x <- seq(h - r, h + r, 0.01)
62       ypositiva <- k + sqrt(r^2 - ((x - h)^2)) # parte positiva de la circunferencia
63       ynegativa <- k - sqrt(r^2 - ((x - h)^2)) # parte negativa de la circunferencia
64       # graficamos primero la parte positiva
65       plot(x, ypositiva, type = "l", xlim = c(h - (r + 1), h + (r + 1)), ylim = c(k - (r + 1), k + (r + 1)),
66           xlab = "Eje X", ylab = "Eje Y")
67       lines(x, ynegativa, type = "l") # agregamos la parte negativa
68       abline(h = 0, v = 0)
69       points(x = h, y = k, col = "red")
70     }
71   } else{
72     return(print("El radio no es positivo."))
73   }
74 }
75
76 # ejecutamos la funcion
77 circunferencia(1, 2, 3)

```

Figura 7: Código de la circunferencia 2

y su gráfica seria:

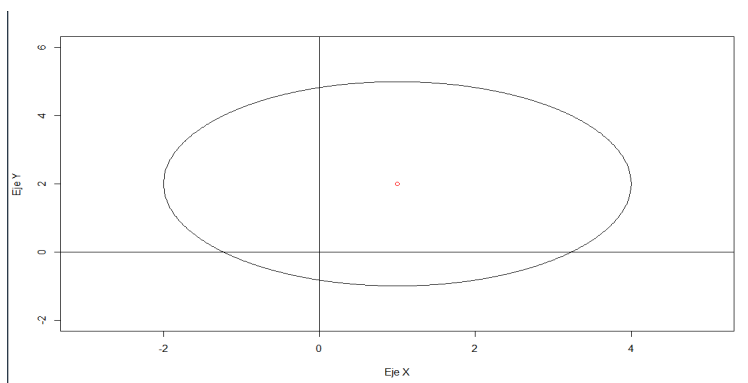


Figura 8: Gráfica de la circunferencia 2

2.3. Parábola

Def. Una parábola es un lugar geométrico de un punto que se mueve en un plano de tal manera que su distancia de una recta fija, situada en el plano, es siempre igual a su distancia de un punto fijo del plano y que no pertenece a la recta.

Las ecuaciones con vértice en el origen

$$y^2 = 4px \quad (4)$$

$$x^2 = 4py \quad (5)$$

Las ecuaciones con vértice fuera del origen

$$(y - k)^2 = 4p(x - h) \quad (6)$$

$$(x - h)^2 = 4p(y - k) \quad (7)$$

Código en R

Para poder graficar una Parábola en R, solo basta con introducir una ecuación que genere una parábola y al igual que en los casos anteriores introducir el código correspondiente para graficar.

Ejemplo 1

```
79 #Parabola 1
80 g <- function(x){
81   return(3*x^2 + x - 6)
82 }
83
84 x <- seq(-6, 6, 0.01) #vector de -6 a 6
85 y <- g(x)
86
87 plot(x, y, type = "l", xlab = "Eje X", ylab = "Eje Y") #graficamos
88 abline(h = 0, v = 0) #una línea horizontal que pasa por el 0 en las x y una l
```

Figura 9: Código de la parábola 1

y su gráfica seria:

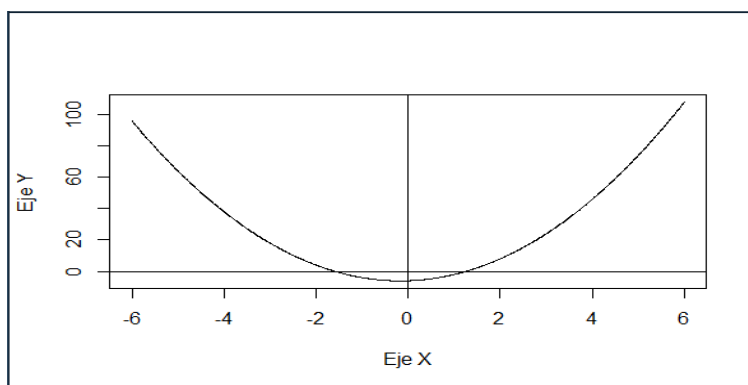


Figura 10: Gráfica de la parábola 1

Ejemplo 2:

```
90 #Parabola 2
91 g <- function(x){
92   return(4*x^2 + x - 8)
93 }
94
95 x <- seq(-8, 8, 0.01)#vector de -5 a 5
96 y <- g(x)
97
98 plot(x, y, type = "l", xlab = "Eje X", ylab = "Eje Y") #graficamos
99 abline(h = 1, v = 1) #una línea horizontal que pasa por el 0 en las x y una
```

Figura 11: Código de la parábola 2

y su gráfica seria:

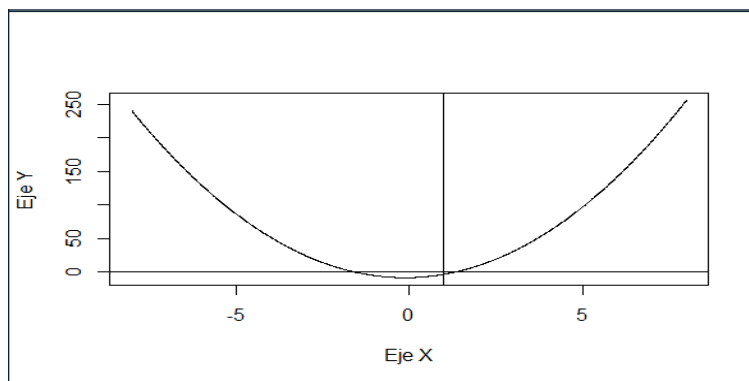


Figura 12: Gráfica de la parábola 2

2.4. Elipse

Def. Una elipse es el lugar geométrico de un punto que se mueve en un plano de tal manera que la suma de sus distancias a dos fijos de ese plano es siempre igual a una constante, mayor que la distancia entre los dos puntos.

Las ecuaciones con vértice en el origen

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (8)$$

$$\frac{x^2}{b^2} + \frac{y^2}{a^2} = 1 \quad (9)$$

Las ecuaciones con vértice fuera del origen

$$\frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2} = 1 \quad (10)$$

$$\frac{(x-h)^2}{b^2} + \frac{(y-k)^2}{a^2} = 1 \quad (11)$$

Código en R

Para poder graficar una elipse en R, se sigue casi el mismo procedimiento que al graficar una circunferencia ya que se gráfica por partes, primero la positiva y luego la negativa, solo que a diferencia de la circunferencia esta tiene muchas condiciones a las cuales esta ligada y se tienen que cumplir. Por ultimo solo es cuestión de asignar valores como el centro, a y b para graficar la elipse

Ejemplo 1

```

101 #Elipse 1
102 elipse <- function(h, k, a, b, horizontal){
103   if (a > b){ # es una elipse horizontal
104     c <- sqrt(a^2 - b^2) # calculamos
105     if (horizontal){ # si es una elipse horizontal
106       x <- seq(h - a, h + a, 0.01) # definimos el dominio
107       ypositiva <- k + sqrt((b^2 - (a^2/a^2) * ((x - h)^2))) # parte positiva
108       ynegativa <- k - sqrt((b^2 - (a^2/a^2) * ((x - h)^2))) # parte negativa
109       # graficamos elipse en partes positivas
110       plot(x, ypositiva, type = "l", xlim = c(h - (a + 1), h + (a + 1)), ylim = c(k - (b + 1), k + (b + 1)),
111            xlab = "Eje X", ylab = "Eje Y")
112       lines(x, ynegativa, type = "l") # graficamos la parte negativa
113       abline(h = 0, v = 0) # eje coordenadas
114       points(x = c(h - c, h + c), y = c(k, k), col = "red") # focos
115     } else {
116       x <- seq(h - b, h + b, 0.01)
117       ypositiva <- k + sqrt((a^2 - (a^2/b^2) * ((x - h)^2)))
118       ynegativa <- k - sqrt((a^2 - (a^2/b^2) * ((x - h)^2)))
119       plot(x, ypositiva, type = "l", xlim = c(h - (b + 1), h + (b + 1)), ylim = c(k - (a + 1), k + (a + 1)),
120            xlab = "Eje X", ylab = "Eje Y")
121       lines(x, ynegativa, type = "l")
122       abline(h = 0, v = 0)
123       points(x = c(h, h), y = c(k - c, k + c), col = "red")
124     }
125   } else {
126     return(print("No cumple las condiciones para ser una elipse. (a no es mayor que b)"))
127   }
128 }
129
130 elipse(0, 0, 9, 4, TRUE)

```

Figura 13: Código de la elipse 1

y su gráfica seria:

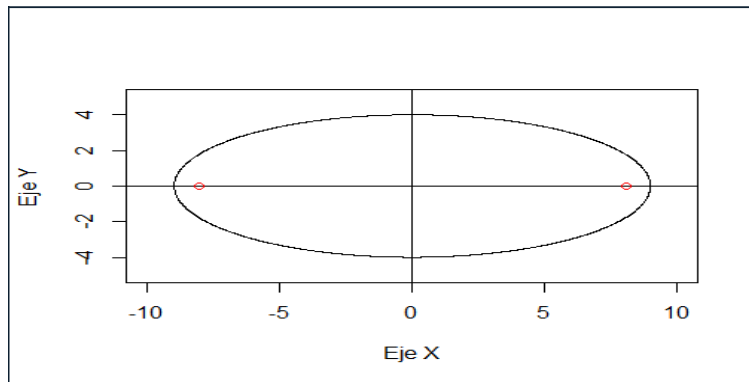


Figura 14: Gráfica de la elipse 1

Ejemplo 2:

```

132 #elipse 2
133 ellipse <- function(h, k, a, b, horizontal){
134   if (a > b){ # es una elipse que sea mayor que b
135     c <- sqrt(a^2 - b^2) # calculamos c
136     if (horizontal){ # si es una elipse horizontal
137       x <- seq(h - a, h + a, 0.01) # definimos el dominio
138       ypositiva <- k + sqrt((b^2 - (b^2/a^2) * ((x - h)^2))) # parte positiva
139       ynegativa <- k - sqrt((b^2 - (b^2/a^2) * ((x - h)^2))) # parte negativa
140       # graficamos primero la parte positiva
141       plot(x, ypositiva, type = "l", xlim = c(h - (a + 1), h + (a + 1)), ylim = c(k - (b + 1), k + (b + 1)),
142         xlab = "Eje X", ylab = "Eje Y")
143       lines(x, ynegativa, type = "l") # agregamos la parte negativa
144       abline(h = 0, v = 0) # eix coordenados
145       points(x = c(h - c, h + c), y = c(k, k), col = "red") # focos
146     } else {
147       x <- seq(h - b, h + b, 0.01)
148       ypositiva <- k + sqrt((a^2 - (a^2/b^2) * ((x - h)^2)))
149       ynegativa <- k - sqrt((a^2 - (a^2/b^2) * ((x - h)^2)))
150       plot(x, ypositiva, type = "l", xlim = c(h - (b + 1), h + (b + 1)), ylim = c(k - (a + 1), k + (a + 1)),
151         xlab = "Eje X", ylab = "Eje Y")
152       lines(x, ynegativa, type = "l")
153       abline(h = 0, v = 0)
154       points(x = c(h, h), y = c(k - c, k + c), col = "red")
155     }
156   } else {
157     return(print("No cumple las condiciones para ser una elipse. (a no es mayor que b)"))
158   }
159 }
160
161 ellipse(1, 1, 3, 1, TRUE)

```

Figura 15: Código de la elipse 2

y su gráfica seria:

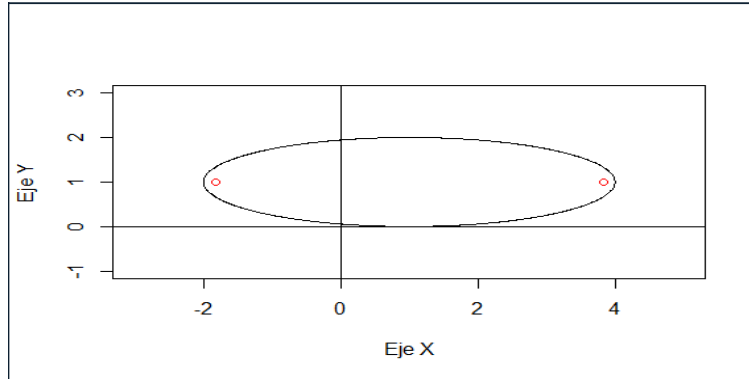


Figura 16: Gráfica de la elipse 2

2.5. Hipérbola

Def. Una hipérbola es el lugar geométrico de un punto que se mueve en un plano de tal manera que el valor absoluto de la diferencia de sus distancias a dos puntos fijos del plano, llamados focos, es siempre igual a una cantidad constante, positiva y menor que la distancia entre los focos

Ecuaciones con vértice en el origen

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1 \quad (12)$$

$$\frac{x^2}{b^2} - \frac{y^2}{a^2} = 1 \quad (13)$$

Ecuaciones con vértice fuera del origen

$$\frac{(x+h)^2}{a^2} - \frac{(y+k)^2}{b^2} = 1 \quad (14)$$

$$\frac{(x+h)^2}{b^2} - \frac{(y+k)^2}{a^2} = 1 \quad (15)$$

Código en R

Para poder graficar una hipérbola en R es necesario crear una función que nos ayude a descifrar como se graficaría nuestra función una vez que se ingresen los valores del centro (h, k) y los valores de a y b, es importante que en la función se agregue ciertas condiciones para que se creen las hipérbolas en ambos lados.

Ejemplo 1:

```

163 #hiperbola 1
164 hiperbola <- function(h, k, a, b, horizontal){
165   c <- sqrt(a^2 + b^2) # calculamos c
166   if (horizontal){ # hiperbola sobre el eje x
167     xizq <- seq(h - (a + 3), h - a, 0.01) # dominio izquierdo
168     xder <- seq(h + a, h + (a + 3), 0.01) # dominio derecho
169     yizqpositiva <- k + sqrt((b^2/a^2)*((xizq - h)^2) - b^2) # parte positiva del dominio izquierdo
170     yizqnegativa <- k - sqrt((b^2/a^2)*((xizq - h)^2) - b^2) # parte negativa del dominio izquierdo
171     yderpositiva <- k + sqrt((b^2/a^2)*((xder - h)^2) - b^2) # parte positiva del dominio derecho
172     ydernegativa <- k - sqrt((b^2/a^2)*((xder - h)^2) - b^2) # parte negativa del dominio derecho
173     # graficamos la parte positiva del dominio izquierdo
174     plot(xizq, yizqpositiva, type = "l", xlim = c(h - (a + 4), h + (a + 4)), ylim = c(k - (b + 4), k + (b + 4)),
175          xlab = "Eje X", ylab = "Eje Y")
176     # resto de la grafica
177     lines(xizq, yizqnegativa, type = "l")
178     lines(xder, ydernegativa, type = "l")
179     lines(xder, yderpositiva, type = "l")
180
181     abline(h = 0, v = 0) # ejes coordenados
182     points(x = c(h - (a + c)), y = c(k), col = "red") # focus

```

Figura 17: Código de la hipérbola 1 parte 1

```

183 } else{ # hiperbola sobre el eje y
184   yizq <- seq(k - (a + 3), k - a, 0.01) # rango inferior
185   yder <- seq(k + a, k + (a + 3), 0.01) # rango superior
186   xizqpositiva <- h + sqrt((b^2/a^2)*((yizq - k)^2) - b^2)
187   xizqnegativa <- h - sqrt((b^2/a^2)*((yizq - k)^2) - b^2)
188   xderpositiva <- h + sqrt((b^2/a^2)*((yder - k)^2) - b^2)
189   xdernegativa <- h - sqrt((b^2/a^2)*((yder - k)^2) - b^2)
190   # graficamos
191   plot(xizqpositiva, yizq, type = "l", xlim = c(h - (b + 4), h + (b + 4)), ylim = c(k - (a + 4), k + (a + 4)),
192        xlab = "Eje X", ylab = "Eje Y")
193   lines(xizqnegativa, yizq, type = "l")
194   lines(xdernegativa, yder, type = "l")
195   lines(xderpositiva, yder, type = "l")
196   abline(h = 1, v = 1)
197   points(x = c(h), y = c(k - (a + c)), col = "red") # focus
198 }
199 }
200
201 hiperbola(5, -3, 4, 8, FALSE)

```

Figura 18: Código de la hipérbola 1 parte 2

y su gráfica seria:

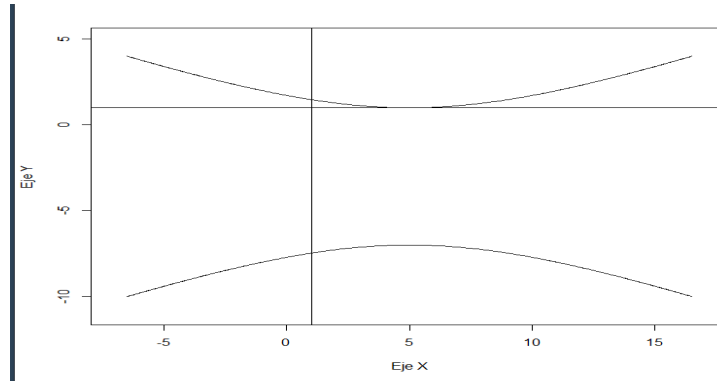


Figura 19: Gráfica de la hipérbola 1

Ejemplo 2:

```

203 #hipérbola 2
204 hipérbola <- function(h, k, a, b, horizontal){
205   c <- sqrt(a^2 + b^2) # calculamos c
206   if (horizontal){ # hipérbola sobre el eje x
207     xizq <- seq(h - (a + 3), h - a, 0.01) # dominio izquierdo
208     xder <- seq(h + a, h + (a + 3), 0.01) # dominio derecho
209     yizqpositiva <- k + sqrt((b^2/a^2)*((xizq - h)^2) - b^2) # parte positiva del dominio izquierdo
210     yizqnegativa <- k - sqrt((b^2/a^2)*((xizq - h)^2) - b^2) # parte negativa del dominio izquierdo
211     yderpositiva <- k + sqrt((b^2/a^2)*((xder - h)^2) - b^2) # parte positiva del dominio derecho
212     ydernegativa <- k - sqrt((b^2/a^2)*((xder - h)^2) - b^2) # parte negativa del dominio derecho
213     # graficamos la parte positiva del dominio izquierdo
214     plot(xizq, yizqpositiva, type = "l", xlim = c(h - (a + 4), h + (a + 4)), ylim = c(k - (b + 4), k + (b + 4)),
215          xlab = "Eje X", ylab = "Eje Y")
216     # resto de la grafica
217     lines(xizq, yizqnegativa, type = "l")
218     lines(xder, ydernegativa, type = "l")
219     lines(xder, yderpositiva, type = "l")
220
221     abline(h = 0, v = 0) # ejes coordenados
222     points(x = c(h - (a + c)), y = c(k), col = "red") # focos
  }
}

```

Figura 20: Código de la hipérbola 2 parte 1

```

223 } else { # hipérbola sobre el eje y
224   yizq <- seq(k - (a + 3), k - a, 0.01) # rango inferior
225   yder <- seq(k + a, k + (a + 3), 0.01) # rango superior
226   xizqpositiva <- h + sqrt((b^2/a^2)*((yizq - k)^2) - b^2)
227   xizqnegativa <- h - sqrt((b^2/a^2)*((yizq - k)^2) - b^2)
228   xderpositiva <- h + sqrt((b^2/a^2)*((yder - k)^2) - b^2)
229   xdernegativa <- h - sqrt((b^2/a^2)*((yder - k)^2) - b^2)
230   # graficamos
231   plot(xizqpositiva, yizq, type = "l", xlim = c(h - (b + 4), h + (b + 4)), ylim = c(k - (a + 4), k + (a + 4)),
232         xlab = "Eje X", ylab = "Eje Y")
233   lines(xizqnegativa, yizq, type = "l")
234   lines(xdernegativa, yder, type = "l")
235   lines(xderpositiva, yder, type = "l")
236   abline(h = 1, v = 1)
237   points(x = c(h), y = c(k - (a + c), col = "red")) # focos
238 }
239 }
240
241 hipérbola(5, 5, 5, 2.64, FALSE)

```

Figura 21: Código de la hipérbola 2 parte 2

y su gráfica seria:

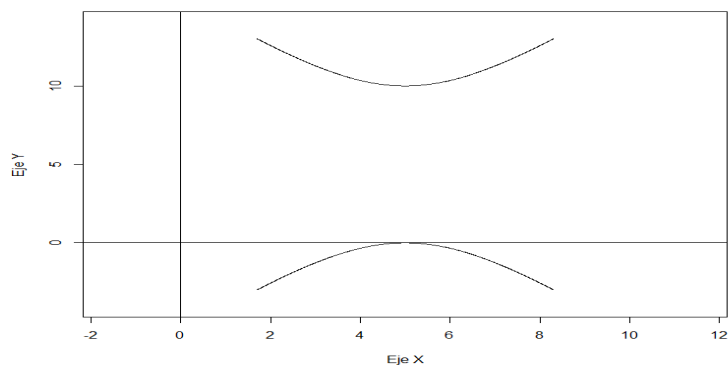


Figura 22: Gráfica de la hipérbola 2

Referencias

- [1] Charles H. Lemhann. Geometria Analitica, 1965.
- [2] Jesus Rangel, repositorio de GitHub <https://github.com/JesusRangel07/MatematicasComputacionales> Repositorio de Github