Matemáticas Computacionales
Practica 1

Cynthia Ivanna Cruz Quiñones
Matricula: 1854499
Grupo: 002

A 20 de Febrero del 2021

Índice

1. Introdcción

En esta primera práctica se hará una de las cosas básicas al momento de aprende 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

[1]**Definición** Llamamos línea recta al lugar geométrico de los puntos tales que tomados dos puntos diferentes cualesquiera $P_1(x_1, y_1)$ y $P_2(x_2, y_2)$ del lugar, el valor de la pendiente m calculado por medio de la fórmula

$$m = \frac{y_1 - y_2}{x_1 - x_2}$$

,tal que $x_1 \neq x_2$, resulta siempre constante.

La formula general es

$$Ax + By + C = 0$$

La ecuación pendiente-ordenada al origen

$$y = mx + b$$

2.1.1. Ejemplos: Linea Recta

```r
    #Ejemplo 1
m <- -2
b <- -1

f <- function(m, b, x){
  return(m * x + b)
}

x <- seq(-2, 2, 0.01)#vector de -2 a 2
y <- f(m, b, x)

plot(x, y, type = "l", xlab = "Eje X", ylab = "Eje Y")
abline(h = 0, v = 0)
```

Cuadro 1: Primer código en R para gráficar la recta de la figura 1.

```r
    #Ejemplo 2
m <- -1/2
b <- 2

f <- function(m, b, x){
  return(m * x + b)
}

x <- seq(-6, 6, 0.01)#vector de -6 a 6
y <- f(m, b, x)

plot(x, y, type = "l", xlab = "Eje X", ylab = "Eje Y")
abline(h = 0, v = 0)
```

Cuadro 2: Segundo código en R para gráficar la recta de la figura 2.

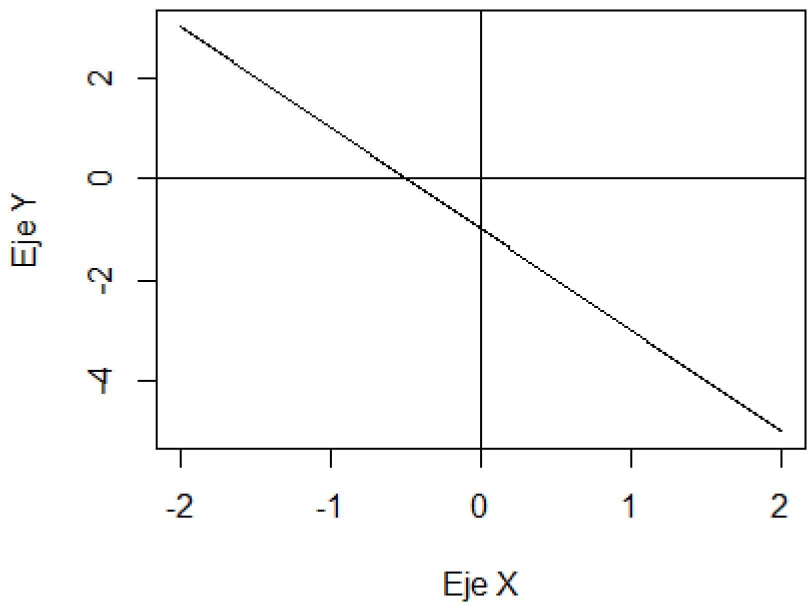

Figura 1: Representación grafica del código

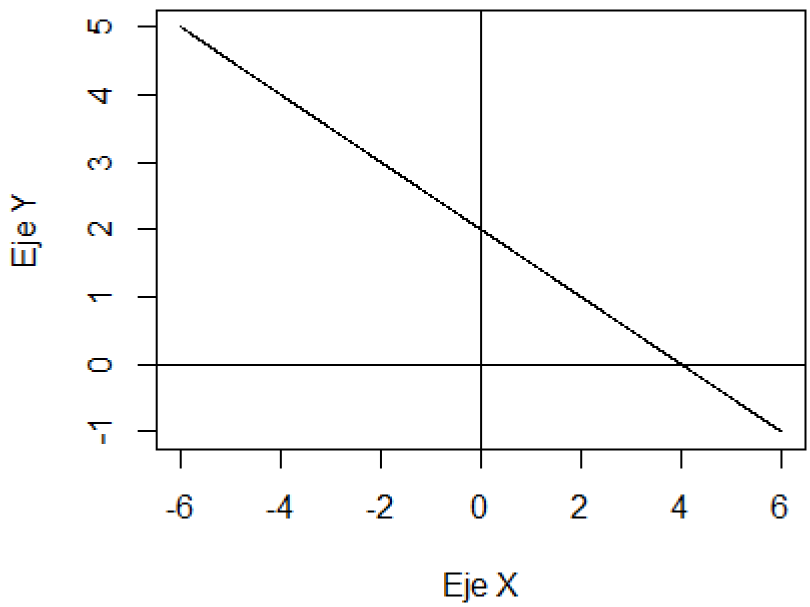

Figura 2: Representacion grafica del codigo

2.2. Parábola

[1]**Definición** Una parábola es el 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 , en siempre igual a la distancia de un punto fijo del plano y que no pertenece a la recta. El punto fijo se llama *foco* y la recta fija *directriz* de la parábola.

La ecuacion de una parábola de vértice en el origen y el eje X, es

$$y^2 = 4px$$

en donde el foco es el punto $(p, 0)$ y la ecuación de la directriz es $x = -p$. Si $p > 0$, la parábola se abre hacia la derecha; si $p < 0$, la parabola se abre hacia la izquierda.

Si el eje de una parábola coincide con el eje Y, el vértice está en el origen, su ecuacĩnon es

$$x^2 = 4py$$

en donde el foco es el punto $(0, p)$, y la ecuación de la directriz es $y = -p$. Si $p > 0$, la parábola se abre hacia arriba; si $p < 0$, la parabola se abre hacia la abajo.

En cada caso, la longitud del lado recto está dada por el valor absoluto de 4p, que es el coeficiente del término de primer grado.

La ecuacion de una parábola de vértice (h, k) y eje paralelo al eje X, es de la forma

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

Si $p > 0$, la parábola se abre hacia la derecha; si $p < 0$, la parábola se abre hacia la izquierda. En cambio, si el vértice es el punto (h, k) y el eje de la parábola es paralelo al eje Y, su ecuación es de la forma

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

Si $p > 0$, la parábola se abre hacia arriba; si $p < 0$, la parábola se abre hacia abajo.

2.2.1. Ejemplos: Parábola

```
#Ejemplo 1

g <- function(x){
  return(1*x^2 + x - 1)
}

x <- seq(-5, 5, 0.01)#vector de -5 a 5
y <- g(x)

plot(x, y, type = "l", xlab = "Eje X", ylab = "Eje Y")
abline(h = 0, v = 0)
```

Cuadro 3: Primer código en R para gráficar la parabola de la figura 3.

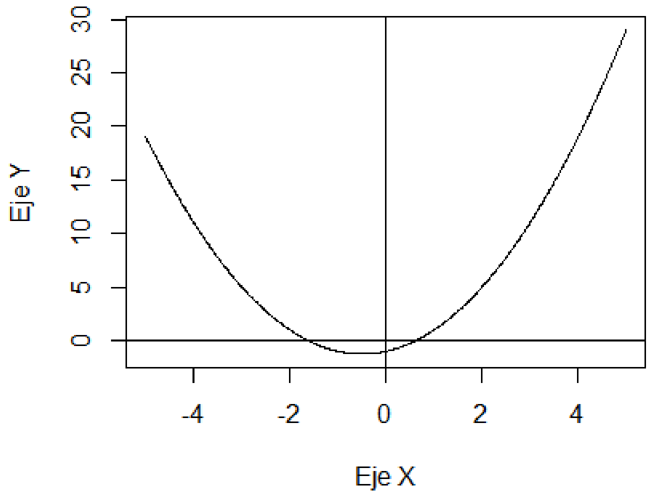

Figura 3: Representación grafica del código

```
#Ejemplo 2

g <- function(x){
  return(4*x^2 +x + 9)
}

x <- seq(-6, 6, 0.01)#vector de -5 a 5
y <- g(x)

plot(x, y, type = "l", xlab = "Eje X", ylab = "Eje Y")
abline(h = 0, v = 0)
```

Cuadro 4: Segundo código en R para gráficar la parabola de la figura 4.

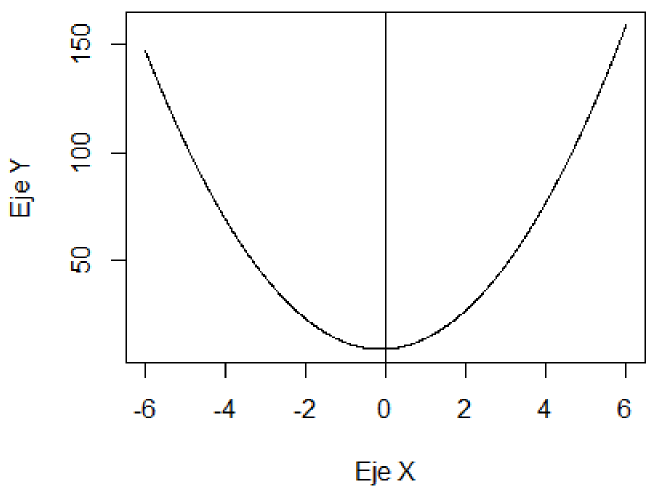

Figura 4: Representación gráfica del codigo

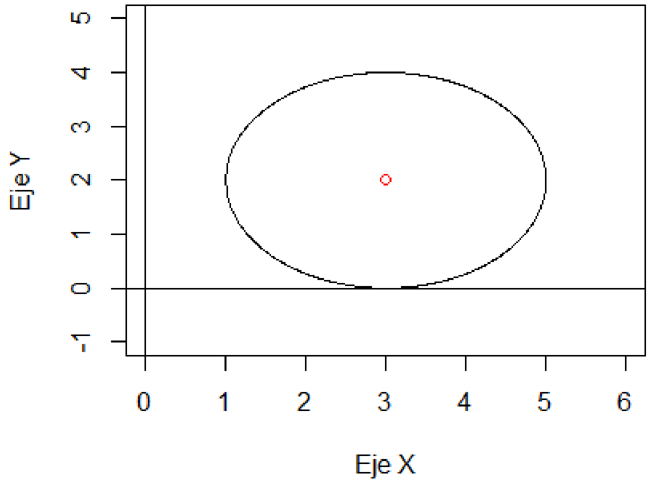

Figura 5: Representación grafica del código

2.3. Circunferencia

[1]**Definición** La circunferencia es el lugar geométrico de un punto que se mueve en el plano de tal manera que se conserva siempre a una constante llamada *radio*

La *circunferencia* cuyo centro es el punto (h,k) y *cuyoradio* es la constante r, tiene por ecuación

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

2.3.1. Ejemplos: Circunferencia

```
#Ejemplo 1

circunferencia1 <- function(h, k, r){
  if (r >= 0){
    if (r == 0){
      plot(x = h, y = k, xlab = "Eje X", ylab = "Eje Y")
    } else{
      x <- seq(h - r, h + r, 0.01)
      ypositiva <- k + sqrt(r^2 - ((x - h)^2))
      ynegativa <- k - sqrt(r^2 - ((x - h)^2))
      plot(x, ypositiva, type = "l", xlim = c(h - (r + 1), h + (r + 1)), ylim = c(k - (r + 1),
           k + (r + 1)),
           xlab = "Eje X", ylab = "Eje Y")
      lines(x, ynegativa, type = "l")
      abline(h = 0, v = 0)
      points(x = h, y = k, col = "red")
    }
  } else{
    return(print("El radio es positivo."))
  }
}

circunferencia(3, 2, 2)
```

Cuadro 5: Primer código en R para gráficar la circunferencia de la figura 5.

```r
#Ejemplo2
circunferencia2 <- function(h, k, r){
  if (r >= 0){
    if (r == 0){
      plot(x = h, y = k, xlab = "Eje X", ylab = "Eje Y") # grafica del punto
    } else{
      x <- seq(h - r, h + r, 0.01)
      ypositiva <- k + sqrt(r^2 - ((x - h)^2))
      ynegativa <- k - sqrt(r^2 - ((x - h)^2))
      plot(x, ypositiva, type = "l", xlim = c(h - (r + 1), h + (r + 1)), ylim = c(k - (r + 1),
           k + (r + 1)),
           xlab = "Eje X", ylab = "Eje Y")
      lines(x, ynegativa, type = "l")
      abline(h = 0, v = 0)
      points(x = h, y = k, col = "red")
    }
  } else{
    return(print("El radio no es positivo."))
  }
}

circunferencia(3, -2, 3)
```

Cuadro 6: Segundo código en R para gráficar la circunferencia de la figura 6.

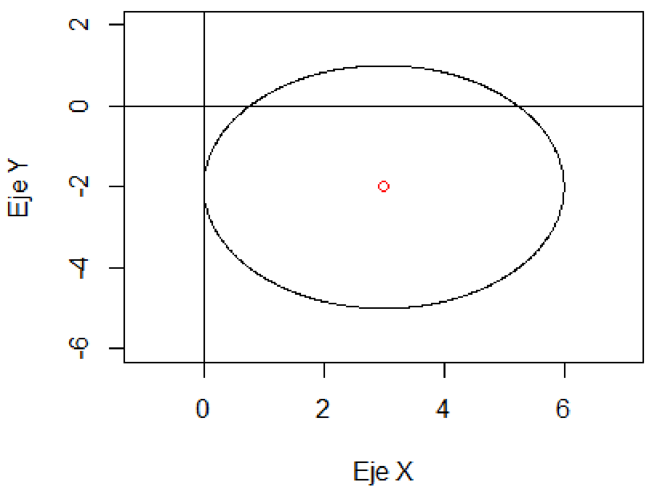

Figura 6: Representación grafica del código

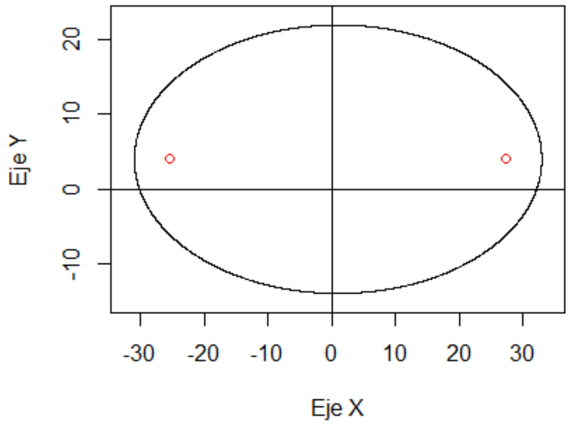

Figura 7: Representación grafica del código

2.4. Elipse

[1]**Definición** 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 puntos fijos de este plano es siempre igual a una constante, mayor que la distancia entre los dos puntos.
Los dos puntos fijos se llaman $focos$ de la elipse.

La ecuación de la elipse de centro en el origen, eje focal coincidente con el $ejeX$, y focos los puntos $(c, 0)$ y $(-c, 0)$, es

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

Si el eje focal coincide con el $ejeY$, de manera que las coordenadas de los focos $(0, c)$ y $(0, -c)$, entonces la ecuación es

$$\frac{y^2}{a^2} + \frac{x^2}{b^2} = 1$$

Ecuacion de la hipérbola con eje transversal *horizontal*

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

Ecuacion de la hipérbola con eje transversal *vertical*

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

Para cada elipse a es la longitud del semejante mayor, b la del semejante menor, c la distancia del centro a cada foco, y a, b, c estan ligadas por la relacion $a^2 = b^2 + c^2$.
También, para cada elipse, la longitud de cada uno de sus lados rectos es $\frac{2b^2}{a}$, y la excentridad e está dada por la relación.

$$e = \frac{c}{a} = \frac{\sqrt{a^2 - b^2}}{a} < 1$$

2.4.1. Ejemplos: Elipse

```r
#Ejemplo 1

elipse1 <- function(h, k, a, b, horizontal){
  if (a > b){
    c <- sqrt(a^2 - b^2)
    if (horizontal){
      x <- seq(h - a, h + a, 0.01)
      ypositiva <- k + sqrt((b^2 - (b^2/a^2) * ((x - h)^2)))
      ynegativa <- k - sqrt((b^2 - (b^2/a^2) * ((x - h)^2)))
      plot(x, ypositiva, type = "l", xlim = c(h - (a + 1), h + (a + 1)), ylim = c(k - (b + 1), k + (b + 1)),
           xlab = "Eje X", ylab = "Eje Y")
      lines(x, ynegativa, type = "l")
      abline(h = 0, v = 0)
      points(x = c(h - c, h + c), y = c(k, k), col = "red")
    } else{
      x <- seq(h - b, h + b, 0.01)
      ypositiva <- k + sqrt((a^2 - (a^2/b^2) * ((x - h)^2)))
      ynegativa <- k - sqrt((a^2 - (a^2/b^2) * ((x - h)^2)))
      plot(x, ypositiva, type = "l", xlim = c(h - (b + 1), h + (b + 1)), ylim = c(k - (a + 1), k + (a + 1)),
           xlab = "Eje X", ylab = "Eje Y")
      lines(x, ynegativa, type = "l")
      abline(h = 0, v = 0)
      points(x = c(h, h), y = c(k - c, k + c), col = "red")
    }
  } else {
    return(print("No cumple las condiciones para ser una elipse. (a no es mayor que b)"))
  }
}

elipse(1, 4, 32, 18, TRUE)
```

Cuadro 7: Primer código en R para gráficar la elipse de la figura 7.

```r
#Ejemplo 2

elipse2 <- function(h, k, a, b, horizontal){
  if (a > b){
    c <- sqrt(a^2 - b^2)
    if (horizontal){
      x <- seq(h - a, h + a, 0.01)
      ypositiva <- k + sqrt((b^2 - (b^2/a^2) * ((x - h)^2)))
      ynegativa <- k - sqrt((b^2 - (b^2/a^2) * ((x - h)^2)))
      plot(x, ypositiva, type = "l", xlim = c(h - (a + 1), h + (a + 1)), ylim = c(k - (b + 1), k + (b + 1)),
           xlab = "Eje X", ylab = "Eje Y")
      lines(x, ynegativa, type = "l")
      abline(h = 0, v = 0)
      points(x = c(h - c, h + c), y = c(k, k), col = "red")
    } else{
      x <- seq(h - b, h + b, 0.01)
      ypositiva <- k + sqrt((a^2 - (a^2/b^2) * ((x - h)^2)))
      ynegativa <- k - sqrt((a^2 - (a^2/b^2) * ((x - h)^2)))
      plot(x, ypositiva, type = "l", xlim = c(h - (b + 1), h + (b + 1)), ylim = c(k - (a + 1), k + (a + 1)),
           xlab = "Eje X", ylab = "Eje Y")
      lines(x, ynegativa, type = "l")
      abline(h = 0, v = 0)
      points(x = c(h, h), y = c(k - c, k + c), col = "red")
    }
  } else {
    return(print("No cumple las condiciones para ser una elipse. (a no es mayor que b)"))
  }
}

elipse(1, 2, 16, -9, TRUE)
```

Cuadro 8: Segundo código en R para gráficar la elipse de la figura 8.

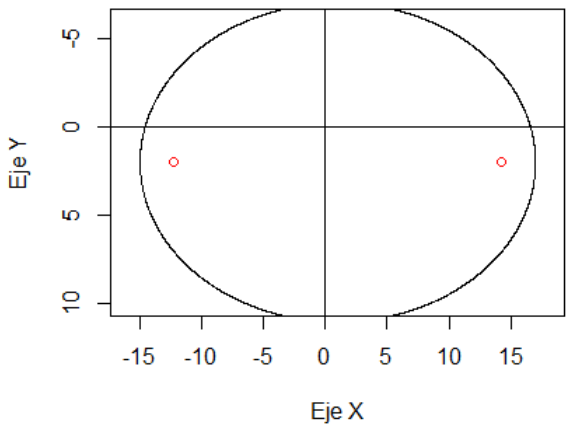

Figura 8: Representación grafica del código

2.5. Hipérbola

[1]**Definición** La 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 sis distancias a dos puntos fijos del plano, llamados *focos*, es siempre iguala una cantidad constante, positiva y menor que la distancia entre los focos.

La ecuación de la hipérbola de centro en el origen, eje focal coincidente con el $ejeX$, y focos los puntos $(c, 0)$ y $(-c, 0)$, es

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$$

Si el eje focal coincide con el $ejeY$, de manera que las coordenadas de los focos $(0, c)$ y $(0, -c)$, entonces la ecuación es

$$\frac{y^2}{a^2} - \frac{x^2}{b^2} = 1$$

Ecuacion de la hipérbola con eje transversal *horizontal*

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

Ecuacion de la hipérbola con eje transversal *vertical*

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

Para cada hipérbola a es la longitud del semejante transverso, b la del semejante conjugado, c la distancia del centro a cada foco, y a, b, c estan ligadas por la relacion $c^2 = a^2 + b^2$. También, para cada hipérbola, la longitud de cada uno de sus lados rectos es $\frac{2b^2}{a}$, y la excentridad e está dada por la relación.

$$e = \frac{c}{a} = \frac{\sqrt{a^2 + b^2}}{a} > 1$$

2.5.1. Ejemplos: Hipérbola

```
#Ejemplo 1

hiperbola2 <- function (h, k, a, b, horizontal ){
  c <- sqrt (a^2 + b^2) #calculamos c
  if( horizontal ){ #hiperbola sobre el eje x
    xizq <- seq(h - (a + 3) , h - a, 0.01) #dominio izquierdo
    xder <- seq(h + a, h + (a + 3) , 0.01) #dominio derecho
    yizqpositiva <- k + sqrt((b^2/a^2) *(( xizq - h) ^2) - b^2) #parte positiva del dominio izquierdo
    yderpositiva <- k + sqrt((b^2/a^2) *(( xder - h) ^2) - b^2)
    yizqnegativa <- k - sqrt((b^2/a^2) *(( xizq - h) ^2) - b^2) #parte negativa del dominio izquierdo
    ydernegativa <- k - sqrt((b^2/a^2) *(( xder - h) ^2) - b^2)
    #graficamos la parte positiva del dominio izquierdo

    plot(xizq , yizqpositiva , type = "l", xlim = c(h - (a + 4) , h + (a + 4)), ylim = c(k - (b + 4) , k + (b + 4)),
         xlab = "Eje X", ylab = "Eje Y")
    #plot(xder , yderpositiva , type = "l",xlim = c(h - (a + 4) , h + (a + 4)), ylim = c(k - (b + 4) , k + (b + 4)))

    lines(xizq , yizqnegativa , type = "l") #agregamos parte negativa del dominio izquierdo
    lines(xder , ydernegativa , type = "l")
    lines(xder , yderpositiva , type = "l")
    lines(xizq , yizqpositiva , type = "l")
    abline(h = 0, v = 0) #ejes coordenados
  } else { #hiperbola sobre el eje y
    yizq <- seq(k - (a + 3) , k - a, 0.01) #rango inferior
    yder <- seq(k + a, k + (a + 3) , 0.01) #rango superior
    xizqpositiva <- h + sqrt((b^2/a^2) *(( yizq - k) ^2) - b^2) #parte positiva del rango inferior
    Xderpositiva <- h + sqrt((b^2/a^2) *(( yder - k) ^2) - b^2)
    xizqnegativa <- h - sqrt((b^2/a^2) *(( yizq - k) ^2) - b^2) #parte negativa del rango superior
    Xdernegativa <- h - sqrt((b^2/a^2) *(( yder - k) ^2) - b^2)
    # graficamos

    plot( xizqpositiva , yizq , type = "l", xlim = c(h - (b + 4) , h + (b + 4)), ylim = c(k - (a + 4) , k + (a + 4)),
          ylab = "Eje Y", xlab = "Eje X")

    lines( xizqnegativa , yizq , type = "l")
    lines( Xdernegativa , yder , type = "l")
    lines( xizqpositiva , yizq , type = "l")
    lines( Xderpositiva , yder , type = "l")
    abline(h = 0, v = 0)
  }
}

hiperbola2 (5, -5, 4, 3, F)
```

Cuadro 9: Primer código en R para gráficar la hipérbola de la figura 9.

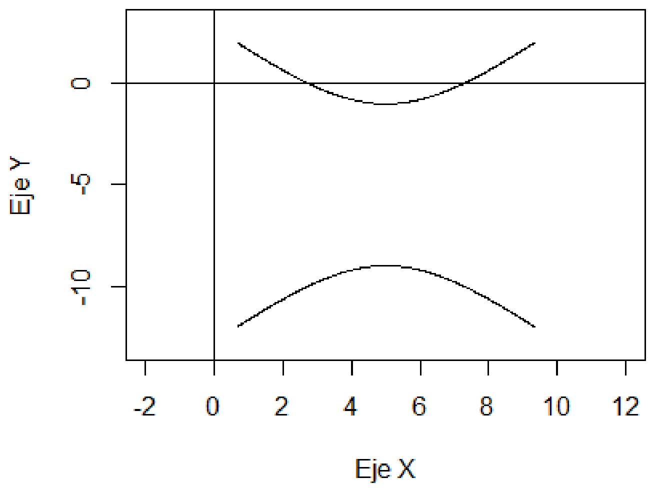

Figura 9: Representación grafica del código

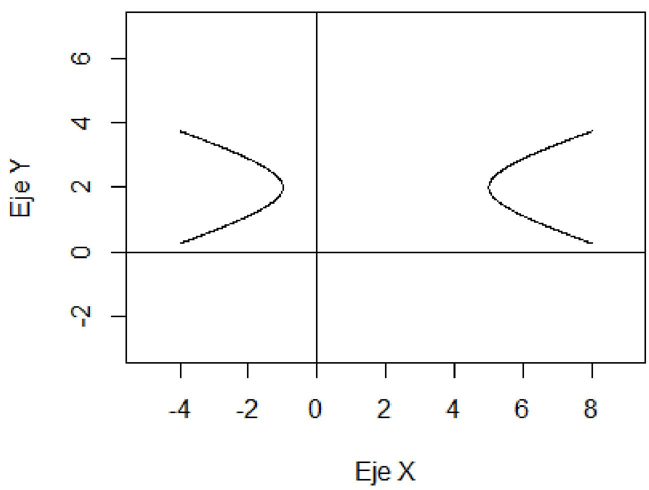

Figura 10: Representación grafica del código

```r
#Ejemplo 2

hiperbola3 <- function (h, k, a, b, horizontal ){
  c <- sqrt (a^2 + b^2) #calculamos c
  if( horizontal ){ #hiperbola sobre el eje x
    xizq <- seq(h - (a + 3) , h - a, 0.01) #dominio izquierdo
    xder <- seq(h + a, h + (a + 3) , 0.01) #dominio derecho
    yizqpositiva <- k + sqrt((b^2/a^2) *(( xizq - h) ^2) - b^2) #parte positiva del dominio izquierdo
    yderpositiva <- k + sqrt((b^2/a^2) *(( xder - h) ^2) - b^2)
    yizqnegativa <- k - sqrt((b^2/a^2) *(( xizq - h) ^2) - b^2) #parte negativa del dominio izquierdo
    ydernegativa <- k - sqrt((b^2/a^2) *(( xder - h) ^2) - b^2)
    #graficamos la parte positiva del dominio izquierdo

    plot(xizq , yizqpositiva , type = "l", xlim = c(h - (a + 4) , h + (a + 4)), ylim = c(k - (b + 4) , k + (b + 4)),
         xlab = "Eje X", ylab = "Eje Y")

    lines(xizq , yizqnegativa , type = "l") #agregamos parte negativa del dominio izquierdo
    lines(xder , ydernegativa , type = "l")
    lines(xder , yderpositiva , type = "l")
    lines(xizq , yizqpositiva , type = "l")
    abline(h = 0, v = 0) #ejes coordenados
  } else { #hiperbola sobre el eje y
    yizq <- seq(k - (a + 3) , k - a, 0.01) #rango inferior
    yder <- seq(k + a, k + (a + 3) , 0.01) #rango superior
    xizqpositiva <- h + sqrt((b^2/a^2) *(( yizq - k) ^2) - b^2) #parte positiva del rango inferior
    Xderpositiva <- h + sqrt((b^2/a^2) *(( yder - k) ^2) - b^2)
    xizqnegativa <- h - sqrt((b^2/a^2) *(( yizq - k) ^2) - b^2) #parte negativa del rango superior
    Xdernegativa <- h - sqrt((b^2/a^2) *(( yder - k) ^2) - b^2)
    # graficamos

    plot( xizqpositiva , yizq , type = "l", xlim = c(h - (b + 4) , h + (b + 4)), ylim = c(k - (a + 4) , k + (a + 4)),
          ylab = "Eje Y", xlab = "Eje X")

    lines( xizqnegativa , yizq , type = "l")
    lines( Xdernegativa , yder , type = "l")
    lines( xizqpositiva , yizq , type = "l")
    lines( Xderpositiva , yder , type = "l")
    abline(h = 0, v = 0)
  }
}

hiperbola3 (2, 2, 3, 1, TRUE)
```

Cuadro 10: Segundo código en R para gráficar la hipérbola de la figura 10.

3. Referencias

[2]Cynthia Ivanna Cruz Quiñones

Referencias

[1] Charles H Lehmann. *Geometría analítica*. LIMUSA, 1965.

[2] Cynthia Ivanna Cruz Quinones. Repositorio de Github. `https://github.com/CynthiaCruzqn/MatematicasComputacionales.git`, 2021.