Método de Newton aplicado a polares

Juan Anagrita, Hector Hernandez, Aldemar Ramirez

Agosto 8,2019

Problema

Hayar la itercepción de dos ecuaciones que estén en coordenadas polares

Solución

Lenguaje de programación: R

Función principal - método de Método de Newton

Parametros:

fun <- función

x0 <- x0 desde donde se comienza la busqueda de la raiz. x0 está en radianes.

tol <- tolerancia minima que debe tener la función

maxiter <- cantidad maxima de iteraciones

Valores de retorno:

x1 <- resultado de la raiz. En radianes.

errorAbsoluto <- vector de errores absolutos de las xn

errorRelativo <- vector de errores relativos de las xn

x <- vector de las xn

```r
library(pracma)

newtonraphson = function(fun, x0, tol, maxiter){

  # f = string
  numiter = 0
  errorAbsoluto = c()
  errorRelativo = c()
  x = c()
  g = parse(text=fun) # parse devuelve tipo "expression"
  g. = D(g,"x")
  fx = function(x){eval(g)} # convertir f a función
  fp = function(x){eval(g.)} # convertir f' a función
  correccion = -fx(x0)/fp(x0)
  while (abs(correccion) >= tol && numiter <= maxiter) {
    numiter = numiter + 1
    if (fp(x0) == 0) stop("División por cero")

    x1 = x0 + correccion
```

```
    x0 = x1
    x <- c(x,x1)
    errorAbsoluto <- c(errorAbsoluto,abs(correccion))
    errorRelativo <- c(errorRelativo,abs(correccion)/(abs(x0)))
    correccion = -fx(x1)/fp(x1)
  }
  if (numiter > maxiter){ warning("Se alcanzó el máximo número de iteraciones.")
    my_list <- list("resultado" = x1, "errorAbsoluto" = errorAbsoluto,
                    "errorRelativo" = errorRelativo, "x" = x)
    return(my_list)


    } else {
    my_list <- list("resultado" = x1, "errorAbsoluto" = errorAbsoluto,
                    "errorRelativo" = errorRelativo, "x" = x)
    return(my_list)
  }
}
```

El método en general nos da la siguiete serie:

$$x_{(n+1)} = x_n - f(x_n)/f'(x_n)$$

Al igual que enc coordenadas rectangulares, el x_0 debe de ser un ángulo cercano al punto de interecepción que se está buscando.

Implementacion

Caso 1: interecepción entre dos gráficas.

A través del método de Newton podemos encontrar en qué ángulo dos ecuaciones se interceptan.

En este caso, ecuación 1:

$$r = 2 + cos(3 * t)$$

Ecuacion 2:

$$r = 2 - e^t$$

Igualando y despejando:

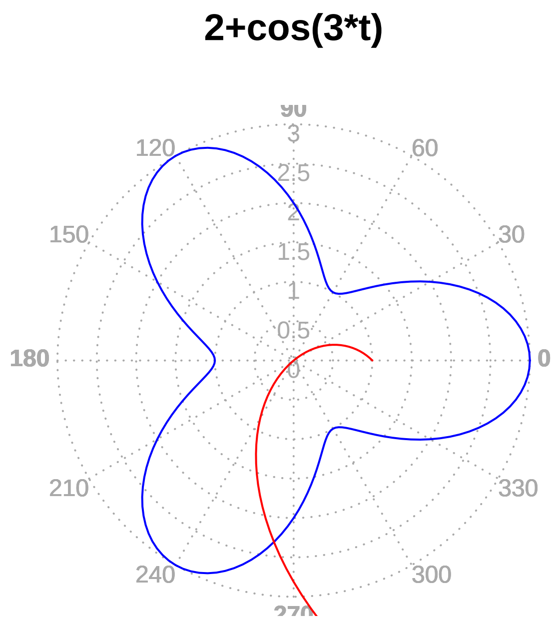
$$f(x) = 2 + cos(3 * t) - (2 - e^t) = 0$$

Graficas

```
t <- deg2rad(seq(0, 360, by = 2))

polar(t, 2+cos(3*t), bxcol = "white", main = "2+cos(3*t)")

polar(t, 2-exp(t), col = "red", add = TRUE)
```




```r
resultados <- newtonraphson("2+cos(3*x) -(2-exp(x))",-1,1e-7,100)

g = parse(text="2+cos(3*x)")

fx = function(x){eval(g)}

cat("Ambas ecuaciones se interceptan en el angulo: ", resultados$resultado, " y en el radio: ", fx(resul
```

```
## Ambas ecuaciones se interceptan en el angulo:  -0.6973291  y en el radio:  1.502087
```

tabla de resultados

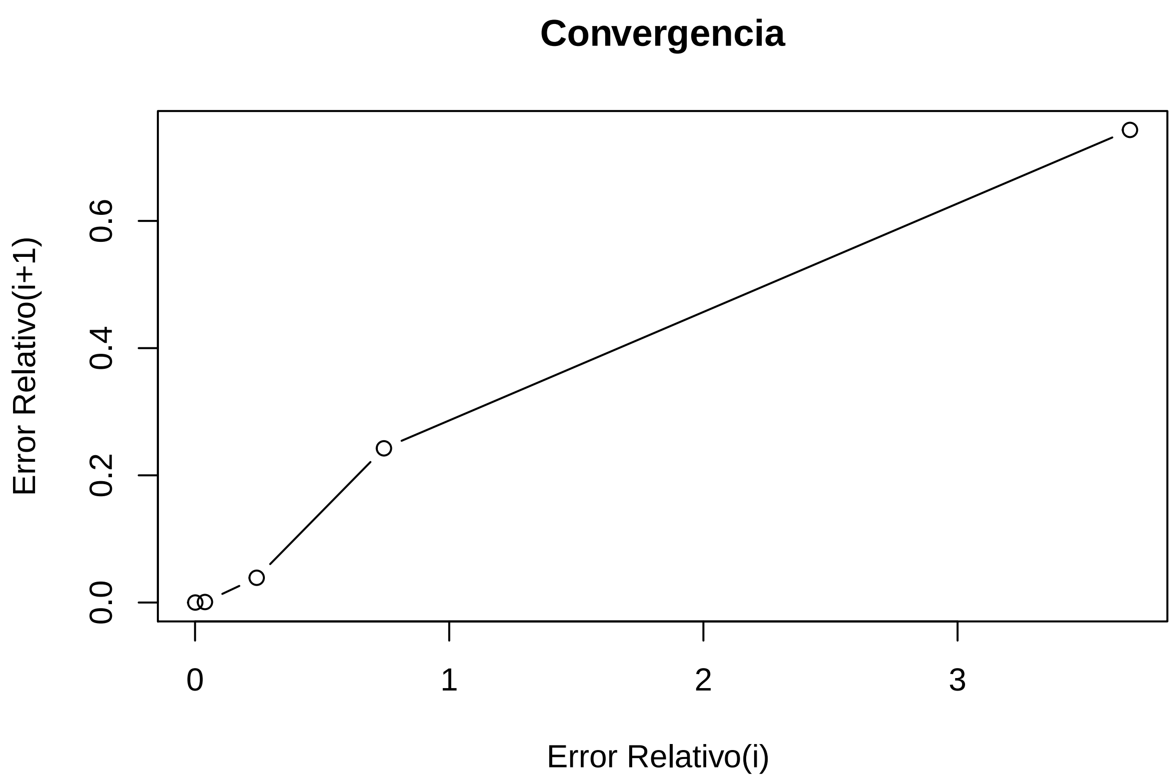
```r
tablaErrores <- data.frame(
"Iteraciones" = 1:length(resultados$errorAbsoluto),
"x_n" = resultados$x,
"Error Absoluto" = resultados$errorAbsoluto,
"Error Relactivo" = resultados$errorRelativo
)
print(tablaErrores)
```

```
##   Iteraciones        x_n Error.Absoluto Error.Relactivo
## 1           1 -0.2137487   7.862513e-01    3.678391e+00
## 2           2 -0.8320496   6.183009e-01    7.431058e-01
```

```
## 3           3 -0.6696807   1.623689e-01    2.424572e-01
## 4           4 -0.6967905   2.710979e-02    3.890666e-02
## 5           5 -0.6973289   5.383876e-04    7.720713e-04
## 6           6 -0.6973291   2.324290e-07    3.333132e-07
```

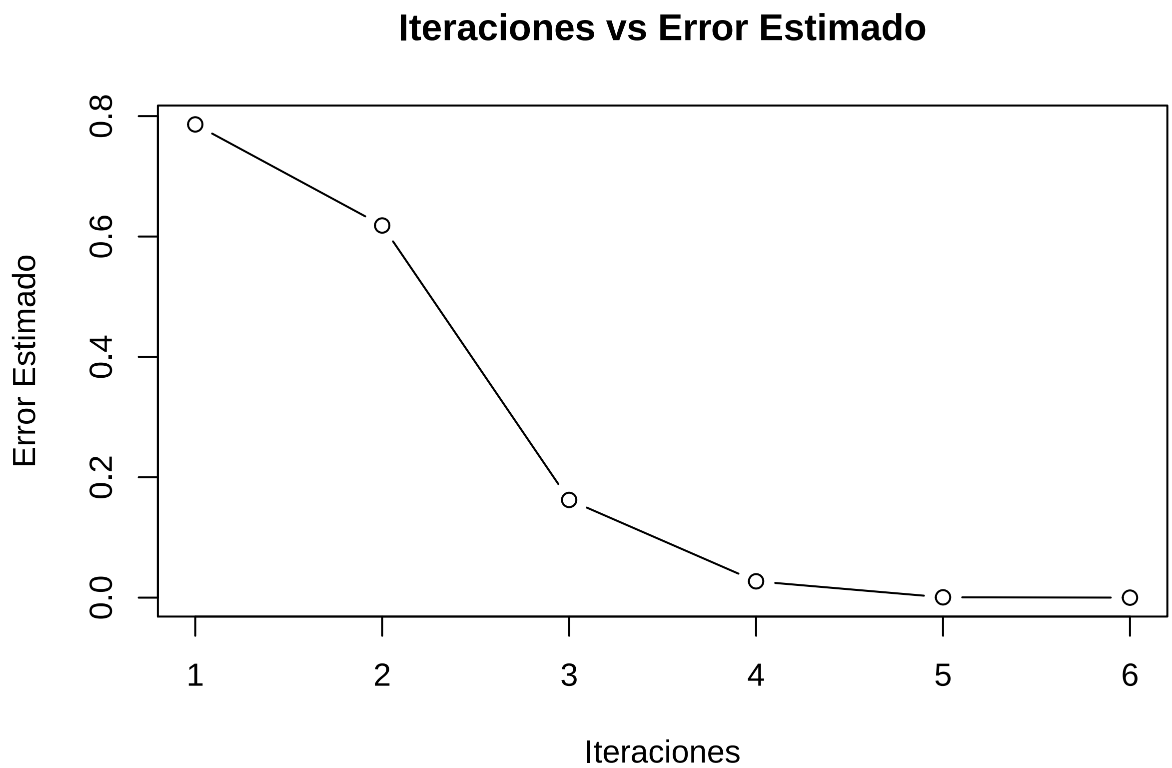
Convergencia

```
m_i = resultados$errorRelativo[-length(resultados$errorRelativo)]
m_i2 = resultados$errorRelativo
m_i2 = m_i2[-1]
plot(x =m_i, y =m_i2, xlab = "Error Relativo(i) ",
ylab = "Error Relativo(i+1)", type="b",main = "Convergencia")
```



El método tiene una convergencia cuadratica.

```
plot(x = 1:length(resultados$x), y = resultados$errorAbsoluto,
xlab = "Iteraciones", ylab = "Error Estimado", type="b",
main = "Iteraciones vs Error Estimado")
```



A través de las iteraciones el error tiende a cero.

Caso 2: En que angulo un ecuación tiene radio = n

Supones que se quiere saber cuando la ecuacion

$$r = cos(x)$$

Tiene radio 0.5. Para este caso se puede interceptar con el circulo de radio 0.5.

$$r = 0.5$$

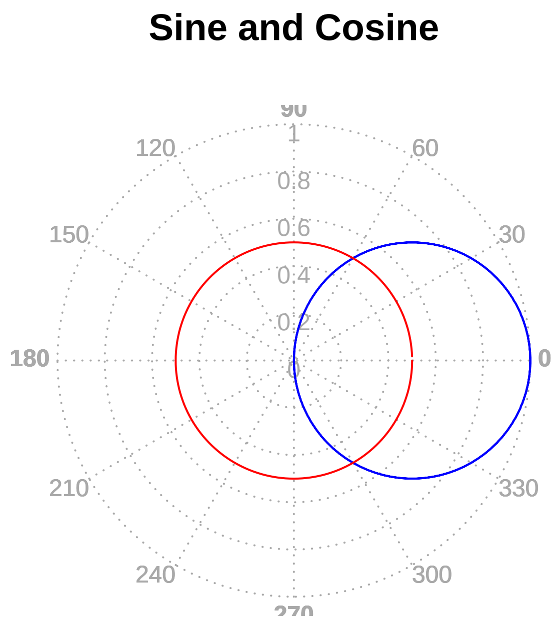
Igualando y despejando ambas ecuaciones:

$$f(x) = cos(x) - 0.5$$

Gráficas

```
polar(t, cos(t), bxcol = "white", main = "Sine and Cosine")

polar(t,0.5*(t/t), col = "red", add = TRUE)
```




```r
resultados <- newtonraphson("cos(x)-0.5",5*pi/4,1e-7,100)

g = parse(text="cos(x)")

fx = function(x){eval(g)}

cat("Ambas ecuaciones se interceptan en el angulo: ", resultados$resultado, " y en el radio: ", fx(resul
```

```
## Ambas ecuaciones se interceptan en el angulo:  5.235988  y en el radio:  0.5
```

```r
cat("5*pi/3 = ", 5*pi/3,"\n")
```

```
## 5*pi/3 =  5.235988
```

Como se puede comprobar $\cos(5.23598776) = \cos(5*pi/3) = 0.5$