Taller Interpolación

Sebastian Angarita, Hector Rodriguez, Aldemar Ramirez

5/9/2019

1. Dados los n + 1 puntos distintos (xi, yi) el polinomio interpolante que incluye a todos los puntos es ´unico

Teorema de Aproximación: Supongase que f este definida y sea continua en [a,b]. Para todo E>0, existe un polinomio P(X) con la propieda de que |f(x)-p(x)|<E para todo x perteneciente al rango [a,b].

Teorema de Unicidad: Sea $x_k n{+}1$ valores distintos(nodos) y sea f una función cuyos valores es esos puntos. Existe un unico P(x) de grado menor o igual a n con identidad: $f(x_k) = P(x_k)$ para todo $k = 0, 1, 2, ..., n$.

Suponemos que hay mas de dos polinomios distintos, P(x) y Q(x) de grado a lo sumo n que verifican $p(x_i) = y$; y $q(x_i) = y$; para $i = 0, 1, ...n$.

Sea el polinomio $r(x) = p(x) - q(x)$ se sabe que para $i = 0, 1..., n$, $r(x_i) = p(x_i) - q(x_i) = 0$. $r(x)$ tendria $n + 1$ raices distintas, ya que x_i es distinto por hipotesis.

Tambien se sabe que $r(x)$ es de grado a lo sumo n, por ser la diferencia de dos polinomios distintos de grado n.

Un polinomio de grado n tiene como maximo n raices distintas o iguales, pero sabemos que $r(x)$ que es de grado a lo sumo n y tiene $n + 1$ raices distintas. Por lo tanto dos raices son iguales y se llega a una contradicción.

2. Construya un polinomio de grado tres que pase por:(0, 10),(1, 15),(2, 5) y que la tangente sea igual a 1 en x0

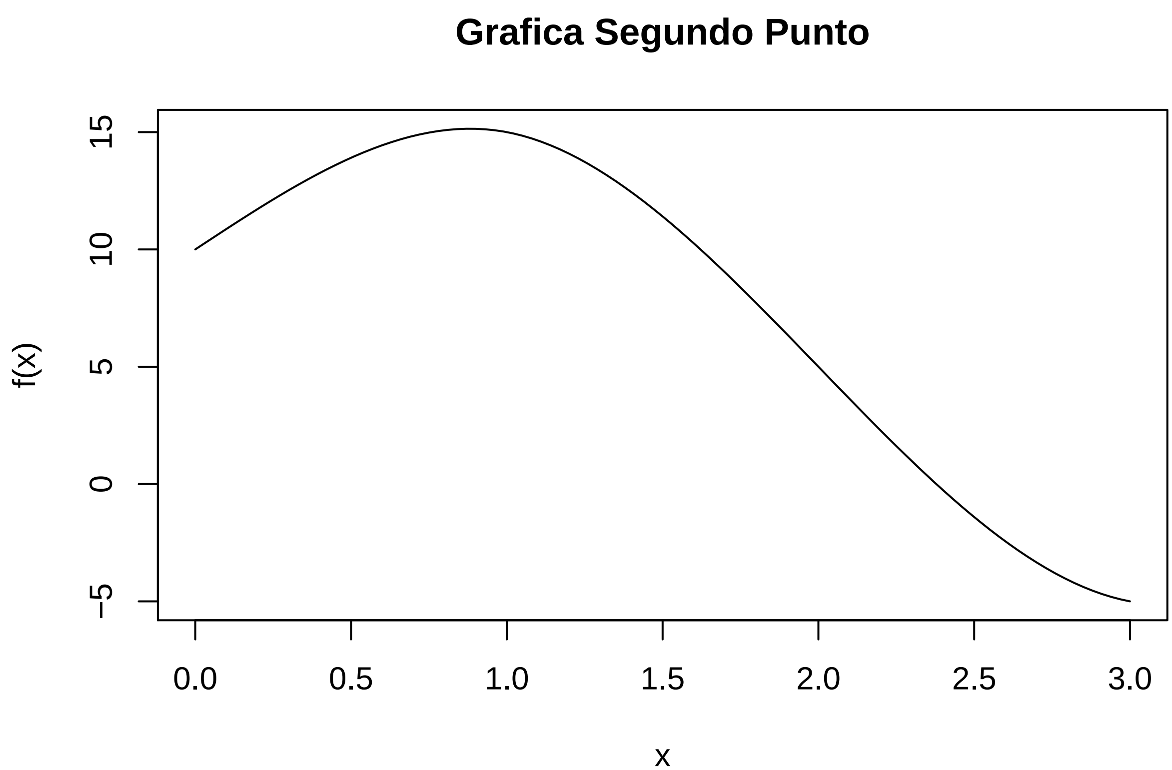
```
require(pracma)
```

```
## Loading required package: pracma
```

```
x = c(0,1,2)
y= c(10,15,5)
xs <- seq(0, 2, by = 1)

pp <- cubicspline(x, y)
ppfun <- function(xs) ppval(pp, xs)

curve(ppfun(x),from=0,to=3,xlab = "x", ylab = "f(x)",
     main = "Grafica Segundo Punto")
```

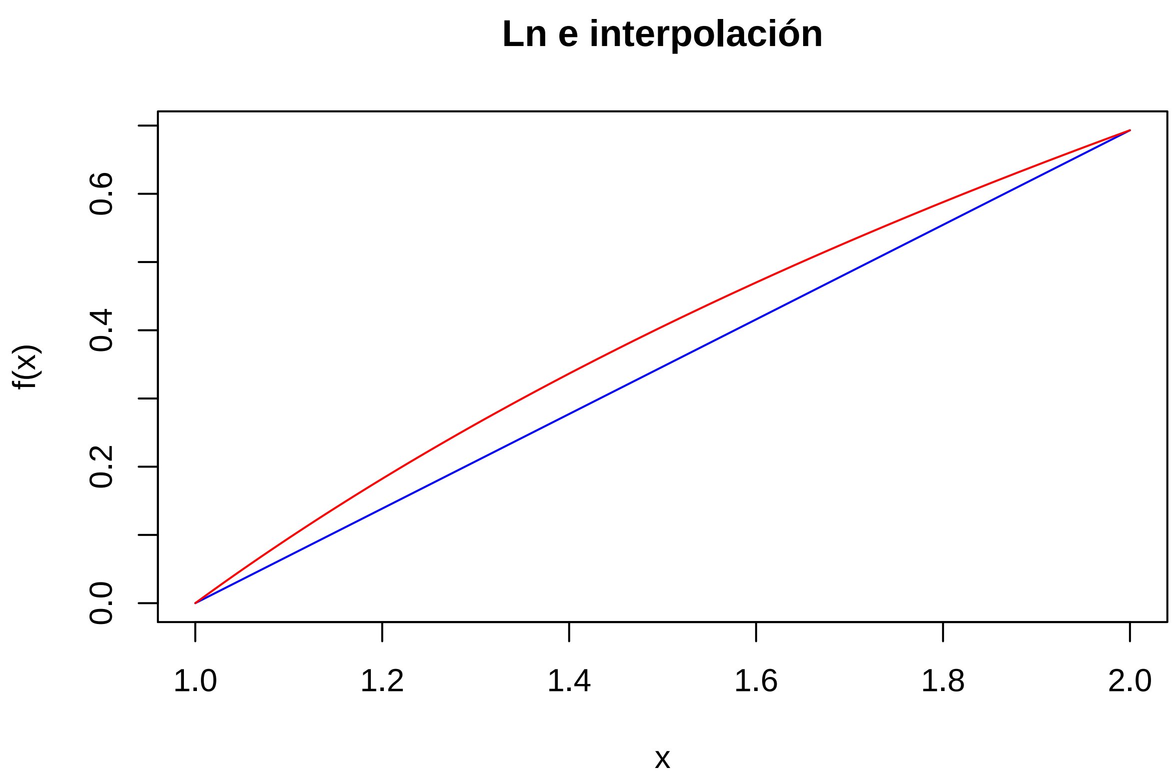


4. Con la función f(x) = lnx construya la interpolación de diferencias divididas en x0 = 1; x1 = 2 y estime el error en [1; 2]

```
newtonInterpolacion = function(x, y, a) {
  n = length(x)
  A = matrix(rep(NA, times = n^2), nrow = n, ncol = n)
  A[,1] = y
  for (k in 2:n) {
    A[k:n, k] = (A[k:n, k-1] - A[(k-1):(n-1), k-1] ) / (x[k:n] - x[1:(n-k+1)])
  }
  # Imprimir matriz de diferencias divididas
  print(A)
  # Evaluar
  smds = rep(NA, length = n)
  smds[1] = 1 #x = x[1],..., x[n] pues n = length(x)
  for (k in 2:n) {
    smds[k] = (a - x[k-1])*smds[k-1] # hasta x[n-1]
  }
  return(sum(diag(A)*smds) )
}
arithmetic.mean <- function(x) {return(sum(x)/length(x))}

f = function(x) log(x)
x1 = c(1,2)
logaritmo = f(x1)
```

```
#Traza de la función Logaritmica
plot(x1,logaritmo,type="l", col="blue",xlab = "x", ylab = "f(x)",
     main = "Ln e interpolación")
#Traza de la recta
curve(log,1,2,add = T, col="red")
```




```
x2 = seq(1,2,by=0.1)
interpolados = c()

for (i in x2) {
  interpolados = c(interpolados,newtonInterpolacion(x1,logaritmo,i));
}
```

```
##           [,1]      [,2]
## [1,] 0.0000000        NA
## [2,] 0.6931472 0.6931472
##           [,1]      [,2]
## [1,] 0.0000000        NA
## [2,] 0.6931472 0.6931472
##           [,1]      [,2]
## [1,] 0.0000000        NA
## [2,] 0.6931472 0.6931472
##           [,1]      [,2]
## [1,] 0.0000000        NA
## [2,] 0.6931472 0.6931472
```

```
##           [,1]      [,2]
## [1,] 0.0000000        NA
## [2,] 0.6931472 0.6931472
##           [,1]      [,2]
## [1,] 0.0000000        NA
## [2,] 0.6931472 0.6931472
##           [,1]      [,2]
## [1,] 0.0000000        NA
## [2,] 0.6931472 0.6931472
##           [,1]      [,2]
## [1,] 0.0000000        NA
## [2,] 0.6931472 0.6931472
##           [,1]      [,2]
## [1,] 0.0000000        NA
## [2,] 0.6931472 0.6931472
##           [,1]      [,2]
## [1,] 0.0000000        NA
## [2,] 0.6931472 0.6931472
##           [,1]      [,2]
## [1,] 0.0000000        NA
## [2,] 0.6931472 0.6931472
```

```r
# Tabla de Interpolados
print(interpolados)
```

```
##  [1] 0.00000000 0.06931472 0.13862944 0.20794415 0.27725887 0.34657359
##  [7] 0.41588831 0.48520303 0.55451774 0.62383246 0.69314718
```

```r
reales = f(x2)
errores = c()

for (i in 1:length(reales)) {
  errores = c(errores,abs(reales[i]-interpolados[i]))
}

# Tabla de errores
print(errores)
```

```
##  [1] 0.00000000 0.02599546 0.04369212 0.05442011 0.05921336 0.05889152
##  [7] 0.05411532 0.04542522 0.03326892 0.01802142 0.00000000
```

```r
#Promedio de error

prom = arithmetic.mean(errores)
print(prom)
```

```
## [1] 0.03573122
```

La traza de color roja hace referencia a la función Ln, mientras que la azul se relaciona con la recta $x_0 = 1; x_1 = 2$.