moodle_2_03-06-13-00

Alan Coila Bustinza

2022-06-03

```r
library(knitr)     # For knitting document and include_graphics function
library(ggplot2)   # For plotting
library('png')
```

pregunta 1

```r
img1_path <- "p1_2022-06-03_130138.png"
include_graphics(img1_path)
```

Calcula el polinomio interpolador de orden 3 de la función $f(x) = 2^x - 3$ usando las abcisas $x = 1,2, 4$ y 5

```r
x1 <- c(1,2,4,5)

f1 <- function(x){
  return(2**x-3)
}

y1 <- f1(x1)
y1
```

```
## [1] -1  1 13 29
```

```r
polyinterp=function(x,y)
{
  if(length(x)!=length(y))
    stop ("La longitud de los vectores x e y debe ser la misma" )
  n=length(x)-1
  vandermonde=rep(1,length(x))
  for(i in 1:n)
  {
    xi=x^i
    vandermonde=cbind(vandermonde,xi)
  }
```

```
  beta=solve(vandermonde,y, tol=1e-22)
  names(beta)=NULL
  return(beta)
}
polyinterp(x1,y1)
```

```
## [1] -4.333333  5.000000 -2.166667  0.500000
```

pregunta 2

```
img1_path <- "p2_2022-06-03_130421.png"
include_graphics(img1_path)
```

En un estudio de fotosíntesis se necesita el area de la hoja de una planta. Midiendo la anchura de la hora en intervalos de $\frac{1}{4}$ cm al largo del eje, el resultado es:

$$\begin{bmatrix} \textit{Distancia} & 0 & \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 & \frac{5}{4} & \frac{3}{2} \\ \textit{Anchura} & 15.890 & 15.942 & 15.993 & 16.044 & 16.094 & 16.144 & 16.193 \end{bmatrix}$$

Usad la regla de Simpson compuesta para encontrar una aproximación del area de la hoja.

```
x2 <- c(0,1/4,1/2,3/4,1,5/4,3/2)
y2 <- c(15.890,15.942,15.993,16.044,16.094,16.144,16.193)



polyinterp=function(x,y)
{
  if(length(x)!=length(y))
    stop ("La longitud de los vectores x e y debe ser la misma" )
  n=length(x)-1
  vandermonde=rep(1,length(x))
  for(i in 1:n)
  {
    xi=x^i
    vandermonde=cbind(vandermonde,xi)
  }
  beta=solve(vandermonde,y, tol=1e-22)
  names(beta)=NULL
  return(beta)
}

horner=function(x,coefs)
```

```
{
  y=rep(0,length(x))
  for(i in length(coefs):1)
    y=coefs[i]+x*y
  return(y)
}

simp_tb <- function(x,y)
{
  m=length(x)-1
  a=x[1]
  b=x[m+1]
  x.ends=x
  y.ends=y
  x.mids=(x.ends[2:(m+1)]-x.ends[1:m])/2+x.ends[1:m]
  y.mids=horner(x.mids, polyinterp(x,y))
  p.area=sum(y.ends[2:(m+1)]+4*y.mids [1:m]+y.ends[1:m])
  p.area=p.area * abs(b - a) / (6 * m)
  return(p.area)
}

simp_tb(x2,y2)
```

```
## [1] 24.06481
```

pregunta 3

```
img1_path <- "p3_2022-06-03_131441.png"
include_graphics(img1_path)
```

Usando el método de Romberg, queremos calcular

$$\int_0^1 \frac{dx}{1+x}.$$

Una parte del esquema triangular toma la siguiente forma:

Nivel 0	*Nivel* 1	...
0.75000000		
0.70833333	0.69444444	
0.69702381	0.69325397	0.69317460
0.69412185	α	0.69314790
0.69339120	0.69314765	0.69314719

Entonces, el valor de α y la aproximación de la integral es:

```r
# primero definir la funcion <- ESTA ES LA PARTE VARIABLE DE CADA
# PREGUNTA
f2 <- function(x){
  return(1/(1+x)) # <--------------------------
}


# luego formula del trapecio compuesta para funcion f
trap=function(f,a,b,m)
{
  x=seq(a,b,length.out=m+1)
  y=f(x)
  p.area=sum((y[2:(m+1)]+y[1:m]))
  p.area=p.area*abs(b-a)/(2*m)
  return(p.area)
}
# para generar el primer nivel 0 ver el numero de valores en nivel 0:

n0 <- function(valores,f,a,b){

  n=c()
  for(i in 2^(0:(valores-1))){
    n=c(n,trap(f2,0,1,i))
  }
  return(n)
}
v0 <- n0(5,f2,0,1)
k=length(v0)
nk <- function(ncero){

  mat <- matrix(0, length(ncero), k)
  mat[,1] <- ncero
  # ((4*n1[i+1]-n1[i])/(4-1))
  # for(i in 2:k){
  #   a=i
  #   b=i+1
  #   mat[,k] <- 4**(k-1)*mat[k,k-1]
  # }
  return (mat)
}
mat <- nk(v0)

for(i in 1:k){
  vect=c()
  for(j in 1:(k-i)){
    if(j>0){
      a=mat[j+1,i]
      b=mat[j,i]
      pot= 4**i
      vect <- c(vect ,(pot*a-b)/(pot-1))
    }
  }
  if(i<k){
```

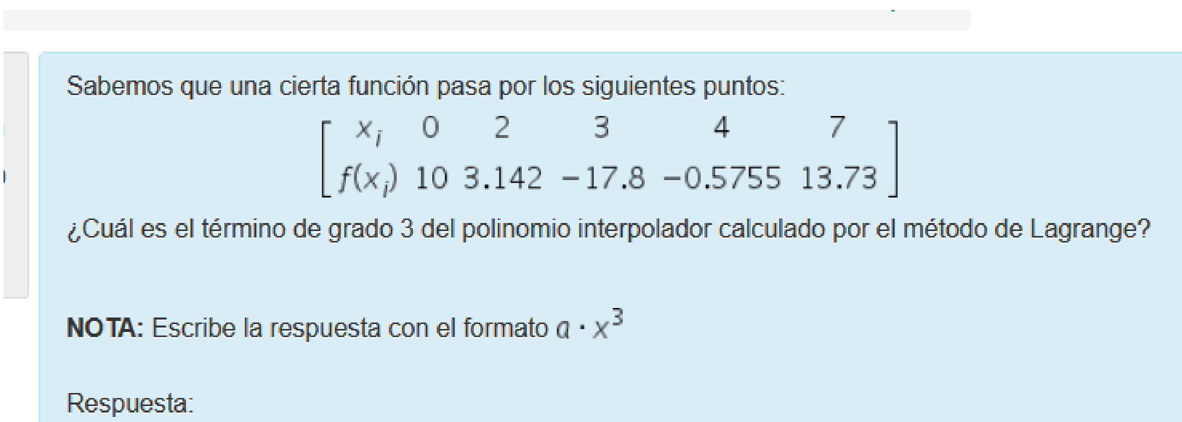
```
    mat[,i+1] <- vect[1:k]
  }
}

mat
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0.7500000 0.6944444 0.6931746 0.6931475 0.6931472
## [2,] 0.7083333 0.6932540 0.6931479 0.6931472        NA
## [3,] 0.6970238 0.6931545 0.6931472        NA        NA
## [4,] 0.6941219 0.6931477        NA        NA        NA
## [5,] 0.6933912        NA        NA        NA        NA
```

pregunta 3

```
img1_path <- "P4_2022-06-03_131852.png"
include_graphics(img1_path)
```


Sabemos que una cierta función pasa por los siguientes puntos:

$$\begin{bmatrix} x_i & 0 & 2 & 3 & 4 & 7 \\ f(x_i) & 10 & 3.142 & -17.8 & -0.5755 & 13.73 \end{bmatrix}$$

¿Cuál es el término de grado 3 del polinomio interpolador calculado por el método de Lagrange?

NOTA: Escribe la respuesta con el formato $a \cdot x^3$

Respuesta:

```
grado=3
x3 <- c(0,2,3,4,7)
y3 <- c(10,3.142,-17.8,-0.5755,13.73)
## rellena todos los valores !!!!!!


Lx <- function(x){
  n <- length(x)
  mult <- function(a,b){
    k <- outer(a,b)
    u <- as.vector(tapply(k, row(k) + col(k), sum))
    return(u)
  }
  mt <- matrix(0,n,n)
  for(i in 1:n){
    f <- c(1)
    m <- c(1)
    for(j in 1:n){
      if(i!=j){
```

```r
        tn <- c(-x[j],1)
        f <- mult(f,tn)

      # }
      # if(j<n+1){

        v <- x[i]-x[j]
        # print(cbind(x[i],x[j]))
        if(v!=0){
          m <- prod(m,v)
        }
      }
    }
    mt[i,] <- y3[i]*f/m # y(x) * Li
    # print('......................')
    # print(f*y3[i])
    # print(m)

  }
  return(mt)
}
r <- Lx(x3) # matriz con cada multiplicacion de f(xi)*Li
r
```

```
##      [,1]        [,2]        [,3]       [,4]        [,5]
## [1,]   10 -12.2619048   5.2976190 -0.9523810  0.05952381
## [2,]    0  13.1964000  -9.5831000  2.1994000 -0.15710000
## [3,]    0  83.0666667 -74.1666667 19.2833333 -1.48333333
## [4,]    0  -1.0071250   0.9831458 -0.2877500  0.02397917
## [5,]    0  -0.7845714   0.8499524 -0.2942143  0.03269048
```

```r
# para hallar el grado tenemos que buscar la columna y sumar sus valores que seran los
# coeficientes de ese grado (variable grado+1 )
sum(r[,grado+1])
```

```
## [1] 19.94839
```