

Práctica 4

En la cuarta práctica debemos aplicar los conocimientos adquiridos en las clases teóricas sobre la interpolación polinómica de Lagrange.

Práctica 4	1
Enunciado	1
Obtener las funciones de base	1
Obtener el polinomio interpolador	2
Ejecutar las funciones	3
Primeros pasos para generar el gráfico	3
Generación del gráfico	4

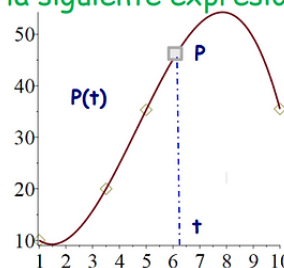
Enunciado

Se conoce la producción de bioetanol que se obtiene en una planta de biocombustibles en función del tiempo. Los valores de la concentración (g/l) están almacenados en el vector: $B(10,20,35,33,35,5)$ y los instantes de tiempo (horas) en el vector $s(1,3,5,10)$.

SE PIDE:

Realizar un script llamado **Bioetanol.R** para estimar, mediante interpolación de Lagrange, la concentración de bioetanol en los instantes $t = 2$ y $t = 6.5$. Para ello, se empleará la siguiente expresión del polinomio interpolador de Lagrange.

$$p = \sum_{i=1}^n B_i L_i$$



Como dice el enunciado, debemos hallar la función del polinomio interpolador. Para ello, lo primero que debemos hacer es hallar las funciones de base de Lagrange.

Obtener las funciones de base

Para ello debemos programar la función que llamaremos "Polbase" para obtener las funciones de base. Para ello, usamos la siguiente expresión.

$$L_i = \prod_{\substack{j=1 \\ j \neq i}}^n \frac{t - s_j}{s_i - s_j}, (i = 1, \dots, n)$$

$i \neq j$

"L" será un vector en cuyas componentes estarán los valores de cada polinomio de base en el punto que estamos usando ("t").

‡ 1) Obtener las funciones de base de la interpolación de Lagrange.

```
Polbase=function(t,s,n){
L=0
for (i in 1:n){
  L[i]=1
  for (j in 1:n){
    if (i!=j){ #en R el no igual se pone !=
      L[i]=L[i]*((t-s[j])/(s[i]-s[j]))
    }
  }
}
return(L)
}
```

Como observamos, "L" va a ser una función que depende del punto "t", de los puntos soporte "s" y del número de puntos soporte "n".

Luego, hacemos un bucle for para el productorio y tenemos en cuenta que i no puede ser igual a j.

Obtener el polinomio interpolador

Lo siguiente será programar la función que nos dará el polinomio interpolador, a la que llamaremos PolInterp. Así obtendremos el polinomio interpolador en el punto "t". Para ello se sigue la fórmula correspondiente.

$$p = \sum_{i=1}^n B_i L_i$$

Así, obtenemos una función que va a depender del vector B, donde van a estar los valores correspondientes a los puntos soporte, y el vector L que acabamos de programar en el apartado anterior.

```
# 2) Obtener el polinomio interpolador en el punto t

Polinterp=function(B,L,n) {
p=0
for (i in 1:n){
    p=p+B[i]*L[i]
    }
return(p)
}
```

Ejecutar las funciones

Una vez creadas las funciones, les asignamos los valores que nos da el enunciado y llamamos a las funciones de vuelta. Lo hacemos para los dos valores de t que nos piden. Las funciones de base se almacenarán en el vector L, y el polinomio interpolador en la variable p1 y p2 respectivamente.

```
#DATOS:
B=c(10,20,35.33,35.5)
s=c(1,3.5,5,10)
t=2
n=length(s)
L=Polbase(t,s,n)
p1=Polinterp(B,L,n)
p1

t=6.5
n=length(s)
L=Polbase(t,s,n)
p2=Polinterp(B,L,n)
p2
```

Primeros pasos para generar el gráfico

Para dibujar el polinomio interpolador, vamos a crear un vector x que tenga 1001 puntos equidistantes en el intervalo [s[1],s[n]]. Por lo tanto, vamos a calcular el valor del polinomio en cada uno de esos puntos.

```
#DIBUJAR LOS PTOS Y POLINOMIO INTERPOLADOR
x=seq(s[1],s[n],length=1001)
```

Como vemos hallar x es muy fácil, ya que lo hacemos con el comando "seq".

Tras esto, llamamos 1001 veces al vector y acumulamos cada uno de los valores en un vector f.

```
f=0
for (k in 1:1001){
    L=Polbase(x[k],s,n)
    f[k]=Polinterp(B,L,n)
}
```

Generación del gráfico

Ya solo queda representar el polinomio interpolador usando el comando "plot". Asignamos límites para los ejes, para que ambas representaciones tengan la misma escala, y ponemos nombres a los ejes.

```
plot(s,B,xlim=c(s[1],s[n]),ylim=c(0,60),xlab='tiempo(h)',ylab='Bioetanol(g/L)');
par(new='true') #Superponer 2 en los mismos ejes.
plot(x,f,xlim=c(s[1],s[n]),ylim=c(0,60),xlab='',ylab='',col='blue',type='l')
```

