

PRÁCTICA 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(1,3,5,5,10).

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.

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

FUNCIONES

Se va a crear una función que partiendo de los datos que tenemos, nos devuelva los polinomios de base de Lagrange (L_i del enunciado). Programar la **función Polbase** para obtener las funciones de base la interpolación de Lagrange, que están dadas por la expresión. Siendo L un vector cuyas componentes son los valores de cada polinomio de base en el punto t:

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

```
Polbase=function(s,t,n){
L=c(0)
for(i in 1:n){
  L[i]=1
  for(j in 1:n){
    if(j!=i){
      L[i]=L[i]*((t-s[j])/(s[i]-s[j]))
    }
  }
}
return(L)
}
```

Más tarde se necesitará asignar L a un nombre para que el programa lo reconozca como existente.

Además, también crearemos otra función con el polinomio interpolador (el dado en el enunciado). La llamaremos PolInterp. Dicha función recibirá como argumentos de entrada: un vector B (que contiene los valores de la función que se interpola), el vector L (que contiene los polinomios de base de Lagrange evaluados en el punto t) la variable n (número de puntos del soporte de interpolación).

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

DATOS

Se deben introducir los datos del enunciado. Además, asignaremos una variable para la llamada de la función, ya que, si en la llamada ponemos function, nos dará error. Si bien, hay que mantener el orden en el que nombramos los datos de los que disponemos: s, t, n.

```

B=c(10,20,35.33,35.5); s=c(1,3.5,5,10); n=length(s)
t=2 #primer caso que nos piden
L=Polbase(s,t,n)
p=PolInterp(B,L,n)
L
p

```

PRODECIMIENTO

Se quiere dibujar conjuntamente los puntos y el polinomio interpolador. Se va a obtener 1011 abscisas equidistantes en el intervalo $[s[1],s[n]]$ y almacenarlas en un vector x . (Calcularemos el valor del polinomio interpolador en cada abscisa para dibujar). Llamamos 1001 veces a las funciones Polbase y PolInterp. La salida de PolInterp se guardará en un vector $f[k]$, $k=1,\dots,1001$.

```

suma=0
x=seq(s[1],s[n],length=1001)
x

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

plot(s,B,xlim=c(s[1],s[n]),xlab="Tiempo(h)",ylab="Concentración
bioetanol(g/l)",ylim=c(0,60))
par(new='TRUE')
plot(x,f,xlim=c(s[1],s[n]),xlab="",ylab="",ylim=c(0,60),col='pink',type='l')

```

El primer gráfico, se utiliza para representar los valores de B en función de las abscisas s , y el segundo para representar los valores de f en función de las abscisas de x .

Establecemos límites en los ejes (el mismo) para que la función quede uniforme, en el segundo gráfico, no se ponen etiquetas en los ejes para que no se superpongan los de un gráfico con el del otro.

