

## Práctica 6 y 7 : Ejercicios sobre todo lo aprendido.

1. Desarrollo del ejercicio 2: Dos métodos de integración numérica: uno determinista y otro estocástico

Se considera una sustancia cuya densidad viene dada por  $d(x)$  siendo  $x$  la coordenada espacial. La masa total en cierto intervalo  $[A,B]$  está dada por

$$M_{AB} = \int_A^B d(x) dx$$

Se desea realizar un programa en R para resolver dicha integral mediante una fórmula de Simpson compuesta, cuya expresión viene dada por:

Solo poner esta fórmula en la function Simpson (A,B,n)

$$\text{Masa} = \int_A^B d(x) dx \approx \frac{h}{6} \left( d(A) + 2 \sum_{i=2}^{n-1} d(T_i) + 4 \sum_{i=1}^{n-1} d\left(\frac{T_i + T_{i+1}}{2}\right) + d(B) \right)$$

Donde se ha realizado una subdivisión del intervalo  $[A,B]$  en  $(n-1)$  subintervalos iguales, es decir, considerando  $n$  puntos  $T_i$ , y siendo  $h$  la longitud de cada subintervalo.

El programa se ejecutará con los siguientes datos:  $A=0$ ,  $B=3.05$ , densidad dada por  $d(x)=\exp(x)*\sin(x)$ . Como número de puntos para el desarrollo de la fórmula se tomará  $n=35$ .

NOTAS: 1. Se programará la expresión de la densidad en una función llamada  $d$ . 2. La fórmula de Simpson compuesta se programará en otra función llamada  $\text{Simpson}$ . 3. Los sumatorios se realizarán con bucles condicionales o secuenciales (como quiera cada uno) 4. El resultado exacto de la masa es:  $M= 11.97907159\dots$

1. Introducimos los datos dados por el enunciado

```
> d=function(x) {  
+   exp(x)*sin(x)  
+ }  
> A=0  
> B=3.05  
> n=35  
> T=seq(A,B,length=n)  
> h=(B-A)/(n-1)
```

2. Componemos la función Simpson (los sumatorios están ya desarrollados)

```

> Simpson=function(A,B,n,T,h){
+   suma1=0
+   suma2=0
+   for(i in 2:(n-1)){
+     suma1=suma1+d(T[i])
+   }
+   for(i in 1:(n-1)){
+     med=(T[i]+T[i+1])/2
+     suma2=suma2+d(med)
+   }
+   I=(h/6)*(d(A)+2*suma1+4*suma2+d(B))
+   return(I)
+ }

```

3. Llamamos a la función y comprobamos el resultado

```

> ValInt=Simpson(A,B,n,T,h)
> ValInt
[1] 11.97907

```

### 3. Método de montecarlo para integración numérica

1. Introducir una variable denominada num\_puntos que contiene el número de valores a considerar (p.ej. 1000 puntos y luego incrementar).

```
40 num_puntos=1000
```

2. Definir extremos del dominio de cálculo: AA=0; BB=7.4 ; A=0; B=3.05 (en nuestro caso). ss =runif(num\_puntos,A,B) ; ff =runif(num\_puntos,AA,BB)

```
41 AA=0; BB=7.4; puntos_dentro=0; kolorr=0; puntos_fuera=0
42 ss=runif(num_puntos,A,B); ff=runif(num_puntos,AA,BB)

```

3. Para i desde 1 hasta num\_puntos

Si (ff[i]<=d(ss[i])) entonces if( ){

    puntos\_dentro = puntos\_dentro+1

    color[i]='blue'

    si no            }else{

        puntos\_fuera = puntos\_fuera+1

        color[i]='orange

    ' Fin condición }

```
43 ▾ for (i in 1:num_puntos){  
44 ▾   if (ff[i]<=d(ss[i])){  
45     puntos_dentro=puntos_dentro+1; kolorr[i]="blue"  
46 ▾   }else{  
47     puntos_fuera=puntos_fuera+1; kolorr[i]="orange"  
48 ▾   }  
49 ▾ }
```

#### 4. Representar conjuntamente ambas zonas (dentro y fuera del área)

```
50 plot(ss,ff,col=kolorr,xlim=c(A,B),ylim=c(AA,BB),xlab="",ylab="")  
51 par(new="true")  
52 plot(x,d(x),type="l",xlab="x",ylab="densidad",xlim=c(A,B),ylim=c(AA,BB))
```