Universidad Rey Juan Carlos

Grado en Matemáticas

Área de una mariposa

Geometría computacional

Autores: Guillermo Grande Santi y

Alejandro López Adrados

Febrero, 2022

Índice

1 Objetivos

Nuestro objetivo en esta práctica es calcular el área de una mariposa descomponiéndola en triángulos. Para ello, presentamos una mariposa cuyo dibujo se extrae de una curva parametrizada. Dicha mariposa de descompondrá en triángulos, y calculando la suma de los áreas de cada uno se obtendrá el resultado buscado.

2 Metodología

2.1 Dibujo de la mariposa

En primer lugar, hemos elegido las ecuaciones paramétricas

$$x = sin(t) * (e^{cos(t)} - 2 * cos(4 * t) - sin(t/12)^5)$$

$$y = cos(t) * (e^{cos(t)} - 2 * cos(4 * t) - sin(t/12)^5)$$

donde $t \in [0, 2\pi]$. Sin embargo, para poder elegir puntos de la mariposa y crear triángulos con facilidad, nos quedaremos con los puntos de la curva con una separación de 0,1 entre cada t. Así, obtenemos 63 puntos que se usarán como vértices de los triángulos. En segundo lugar, se agruparán los vértices en tripletes según los triángulos que queremos (en total han sido 58) y se meterán en una matriz. Tras esto, se recorre la matriz con un bucle que irá pintando cada triángulo en el plano xy con la función polygon (ver anexo).

2.2 Área de la mariposa

Para calcular el área de cada triángulo teniendo sus tres vértices, hemos creado una función "área" en la que se hará el determinante de dichos vértices puestos en filas (añadiendo un 1 en la tercera coordenada de cada uno) multiplicado por $\frac{1}{2}$ y devolviéndolo en valor absoluto.

$$\acute{A}rea = \frac{1}{2} \cdot \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$$

Dentro del bucle comentado anteriormente, se irá llamando a esta función por cada fila de la matriz (por cada triángulo) y ser irá acumulando el área. Finalmente, se obtendrá la suma de todas las áreas y se imprimirá por pantalla.

3 Conclusiones

La realización de esta práctica ha sido muy tediosa debido a que hemos introducido cada triángulo manualmente. Es probable que haya otra forma de dibujar la mariposa más automatizada pero ésta es la que nos daba mejor resultado. Por otra parte, hemos realizado esta memoria en LaTeX y hemos aprendido a indexar código de R en ella. Es por ello que ha sido una práctica muy satisfactoria para nosotros y pensamos que el resultado es bueno.

4 Anexos

4.1 Código, Dibujo y Resultado

```
t<-seq(0,2*pi,0.1)
x<-sin(t)*(exp(1)^cos(t)-2*cos(4*t)-sin(t/12)^5)
y<-cos(t)*(exp(1)^cos(t)-2*cos(4*t)-sin(t/12)^5)
plot(-4:4,-4:4)
areaTotal<-0
#polygon(x,y, col = "white", border = "white")

area<-function(z){
  v<-c(x[z[1]],y[z[1]],1)
```

```
  w<-c(x[z[2]],y[z[2]],1)
  u<-c(x[z[3]],y[z[3]],1)
  matriz<-rbind(v,w,u)
  area<-det(matriz)
  area<-abs(area)
  area<-area/2
}

#ALA DERECHA
#v1<-c(1, 34, 35)
v2<-c(3, 35, 14)
v3<-c(29, 35, 14)
v4<-c(7, 6, 5)
v5<-c(8, 7, 5)
v6<-c(9, 8, 5)
v7<-c(10, 9, 5)
v8<-c(11, 10, 5)
v9<-c(12, 11, 5)
v10<-c(13, 12, 5)
v11<-c(14, 13, 5)
v12<-c(4, 14, 5)
v13<-c(3, 14, 4)

#ALA IZQUIERDA
v14<-c(52, 60, 51)
v15<-c(60, 53, 52)
v16<-c(60, 54, 53)
v17<-c(60, 55, 56)
v18<-c(60, 55, 54)
v19<-c(60, 55, 56)
```

```r
v20<-c(60, 57, 56)
v21<-c(60, 57, 58)
v22<-c(60, 59, 58)
v23<-c(61, 60, 51)
v24<-c(61, 62, 51)
v25<-c(30, 62, 51)
v26<-c(30, 29, 51)
v27<-c(30, 62, 31)

#CABEZA
v28<-c(31, 33, 32)
v29<-c(1, 2, 3)
v30<-c(3, 1, 33)
v31<-c(31, 33, 1)
v32<-c(31, 63, 62)
v33<-c(31, 63, 1)

#MINI ALA IZQUIERDA
v34<-c(15, 16, 17)
v35<-c(17, 18, 19)
v36<-c(15, 19, 17)
v37<-c(15, 19, 29)

#MINI ALA DERECHA
v38<-c(50, 49, 48)
v39<-c(48, 47, 46)
v40<-c(46, 50, 48)
v41<-c(50, 29, 46)

#ALA INFERIOR IZQUIERDA
```

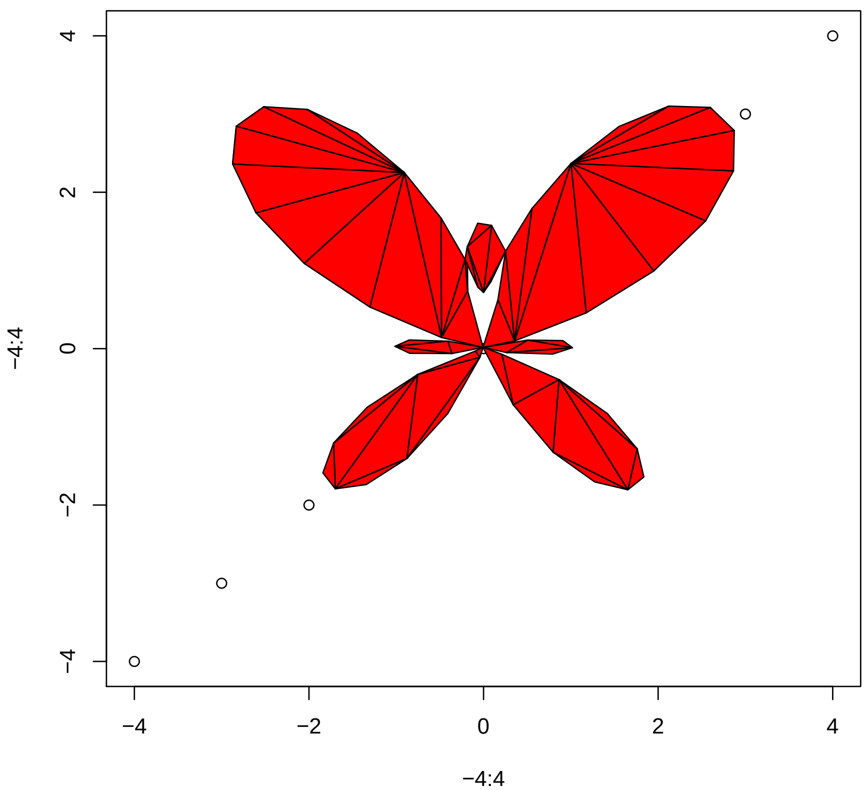
```r
v42<-c(40, 42, 44)
v43<-c(40, 42, 44)
v44<-c(40, 38, 44)
v45<-c(38, 44, 36)
v46<-c(42, 43, 44)
v47<-c(38, 39, 40)
v48<-c(38, 37, 36)
v49<-c(40, 41, 42)
v50<-c(36, 45, 44)
v51<-c(36, 45, 29)

#ALA INFERIOR DERECHA
v52<-c(25, 24, 23)
v53<-c(21, 23, 25)
v54<-c(21, 23, 22)
v55<-c(25, 26, 27)
v56<-c(21, 25, 27)
v57<-c(21, 27, 28)
v58<-c(20, 21, 28)
v59<-c(28, 29, 20)

#Introducimos todos los vectores en una matriz
vectores<-rbind(v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,
                v16,v17,v18,v19,v20,v21,v22,v23,v24,v25,v26,v27,
                v28,v29,v30,v31,v32,v33,v34,v35,v36,v37,v38,v39,
                v40,v41,v42,v43,v44,v45,v46,v47,v48,v49,v50,v51,
                v52,v53,v54,v55,v56,v57,v58,v59)

#Dibujamos cada triángulo formado por los vértices y calculamos
#su área
```

```
for(i in 1:58){
  polygon(c(x[vectores[i,1]],x[vectores[i,2]],x[vectores[i,3]]),
  c(y[vectores[i,1]],y[vectores[i,2]],y[vectores[i,3]]),
  col="red")
  areaTotal<-area(vectores[i,])+areaTotal
}
```




```
areaTotal
```

```
## [1] 13.24793
```