

CÓDIGO

```
cat("\014")
rm(list=ls())
#Crea un vector que simbolice la matriz de cambio A y le pide al usuario que la pueble con valores
Matriz_de_cambio=matrix(0,nrow=2,ncol=2)
for (i in 1:2) {
  for (j in 1:2) {
    Matriz_de_cambio[i,j]=as.numeric(readline(cat("Inserta el valor de la matriz A [",i,";",j,""])))
  }
}
#Aqui me facilito la vida, llamando a los valores ABCD para facilitarme escribir los cálculos
A=Matriz_de_cambio[1,1]
B=Matriz_de_cambio[1,2]
C=Matriz_de_cambio[2,1]
D=Matriz_de_cambio[2,2]
#Aqui calculo los autovalores, insertando en la ecuación cuadrática
Autovalor_primeroysegundo=(A+D+sqrt((A+D)^2-(4*(A*D-B*C))))/2
Autovalor_segundoyprimero=(A+D-sqrt((A+D)^2-(4*(A*D-B*C))))/2
vector_primeroysegundo=numeric(2)
vector_primeroysegundo[1]=1
vector_segundoyprimero=numeric(2)
vector_segundoyprimero[1]=1
vector_primeroysegundo[2]=-((A-Autovalor_primeroysegundo)/B)
vector_segundoyprimero[2]=-((A-Autovalor_segundoyprimero)/B)
N=as.numeric(readline(cat("Inserta el número de puntos en x o y de inicio que quieres que tome")))
x_o=seq(-25, 25, length.out=N)
y_o=seq(-25, 25, length.out=N)
plot(NA, xlim = c(-25, 25), ylim = c(-25, 25),
```

```

xlab = "Eje X", ylab = "Eje Y",
main = "Gráfica del sistema")
for (i in 1:N) {
  for (j in 1:N) {
    c_segundo=(((Autovalor_primer-A)*x_o[i])-(B*y_o[j]))/(Autovalor_primer+Autovalor_segundo)
    c_primer=x_o[i]-c_segundo

x_fin=c_primer*((exp(1))^(0.5*Autovalor_primer))+c_segundo*((exp(1))^(0.5*Autovalor_segundo)
)
  y_fin=(((Autovalor_primer-
A)/B)*c_primer*((exp(1))^(0.5*Autovalor_primer)))+(((Autovalor_segundo-
A)/B)*c_segundo*((exp(1))^(0.5*Autovalor_segundo)))
  arrows(x_o[i],y_o[j],x_fin,y_fin, col="black", length=0.05)
}
}

```

EXPLICACIÓN

1. Limpieza del entorno

```
cat("\014")
```

```
rm(list=ls())
```

El comando `cat("\014")` limpia la consola, similar a la función de "limpiar consola".

El comando `rm(list=ls())` elimina todas las variables, objetos y funciones que estén almacenados en la memoria de R.

2. Creación e inicialización de la matriz de cambio

```
Matriz_de_cambio = matrix(0, nrow=2, ncol=2)
```

```
for (i in 1:2){  
  for (j in 1:2){  
    Matriz_de_cambio[i, j] = as.numeric(readline(cat("Inserta el valor de la matriz A [", i, ";", j, "]: ")))  
  }  
}
```

Se crea una matriz de 2x2 inicializada con ceros.

El doble bucle for solicita al usuario que ingrese los valores de la matriz de cambio, uno por uno, y los asigna a las posiciones correspondientes de la matriz.

La función readline() captura la entrada del usuario y as.numeric() convierte la entrada a un número.

3. Asignación de variables a partir de la matriz

```
A = Matriz_de_cambio[1, 1]
```

```
B = Matriz_de_cambio[1, 2]
```

```
C = Matriz_de_cambio[2, 1]
```

```
D = Matriz_de_cambio[2, 2]
```

Se asignan los elementos de la matriz de cambio a las variables A, B, C y D para facilitar los cálculos posteriores.

A es la posición (1,1), B es la posición (1,2), C es la posición (2,1) y D es la posición (2,2) de la matriz.

4. Cálculo de los autovalores

```
Autovalor_primer = (A + D + sqrt((A + D)^2 - (4 * (A * D - B * C)))) / 2
```

```
Autovalor_segundo = (A + D - sqrt((A + D)^2 - (4 * (A * D - B * C)))) / 2
```

Se calculan los autovalores de la matriz usando la fórmula de la ecuación cuadrática.

La fórmula es la misma que la fórmula general para resolver ecuaciones cuadráticas:

$$\lambda = \frac{A + D \pm \sqrt{(A + D)^2 - 4(AD - BC)}}{2}$$

Se obtienen dos autovalores, Autovalor_primeros y Autovalor_segundo.

5. Cálculo de los autovectores (vectores propios)

```
vector_primeros = numeric(2)
```

```
vector_primeros[1] = 1
```

```
vector_segundo = numeric(2)
```

```
vector_segundo[1] = 1
```

```
vector_primeros[2] = -((A - Autovalor_primeros) / B)
```

```
vector_segundo[2] = -((A - Autovalor_segundo) / B)
```

Se calculan los autovectores correspondientes a los autovalores.

Para obtener los autovectores, se usa la relación $(A - \lambda)x + By = 0$.

Para simplificar, se toma $x = 1$, y se resuelve para y .

El resultado se almacena en los vectores `vector_primeros` y `vector_segundo`.

6. Entrada de puntos iniciales y creación de la cuadrícula

```
N = as.numeric(readline(cat("Inserta el número de puntos en x o y de inicio que quieres que tome:
")))

```

```
x_o = seq(-25, 25, length.out = N)

```

```
y_o = seq(-25, 25, length.out = N)

```

Se pide al usuario que ingrese el número de puntos N que quiere usar para la cuadrícula inicial.

Se generan N puntos igualmente espaciados en el intervalo de -25 a 25 para los ejes x e y .

El tamaño de la cuadrícula será $N \times N$ puntos.

7. Configuración del espacio de la gráfica

```
plot(NA, xlim = c(-25, 25), ylim = c(-25, 25),  
     xlab = "Eje X", ylab = "Eje Y",  
     main = "Gráfica del sistema")
```

Se crea un gráfico vacío con los ejes x e y en el intervalo de -25 a 25.

El título de la gráfica se establece como "Gráfica del sistema".

Los nombres de los ejes se definen como "Eje X" y "Eje Y".

8. Cálculo de la dinámica y dibujo de las flechas

```
for (i in 1:N) {  
  for (j in 1:N) {  
    c_segundo = (((Autovalor_primeros - A) * x_o[i]) - (B * y_o[j])) / (Autovalor_primeros +  
Autovalor_segundos))  
    c_primeros = x_o[i] - c_segundo  
    x_fin = c_primeros * ((exp(1))^(0.5 * Autovalor_primeros)) + c_segundo * ((exp(1))^(0.5 *  
Autovalor_segundos))  
    y_fin = (((Autovalor_primeros - A) / B) * c_primeros * ((exp(1))^(0.5 * Autovalor_primeros))) +  
            (((Autovalor_segundos - A) / B) * c_segundo * ((exp(1))^(0.5 * Autovalor_segundos)))  
    arrows(x_o[i], y_o[j], x_fin, y_fin, col = "black", length = 0.05)  
  }  
}
```

Se usa un doble bucle for para recorrer la cuadrícula de puntos generada con x_o y y_o .

En cada punto de la cuadrícula, se calculan c_{primeros} y c_{segundos} , que son los coeficientes de la combinación lineal de los autovectores.

x_{fin} y y_{fin} se calculan a partir de la evolución de cada punto inicial según la matriz de cambio.

Finalmente, se dibuja una flecha desde el punto $(x_o[i], y_o[j])$ hasta el punto (x_{fin}, y_{fin}) con la función `arrows()`.

La flecha se dibuja en negro con un tamaño de punta de flecha de 0.05.