

## Uso de método de bases de Lagrange para interpolación polinómica en R

En primer lugar, realizamos una función (`lagrange_coefs`) que calcula los coeficientes de los polinomios base de Lagrange. Toma como entrada los puntos de soporte y devuelve una matriz donde cada fila representa los coeficientes de un polinomio base de Lagrange. Además, hay que crear una función (`imprime_bases`) que toma la matriz de coeficientes de las bases de Lagrange y las imprime en un formato legible. Cada base se muestra como un polinomio, con los coeficientes redondeados a cuatro decimales.

```
#Función para calcular los coeficientes de Lagrange
```

```
lagrange_coefs <- function(soporte) {  
  n <- length(soporte) - 1  
  coefs <- matrix(0, nrow = n + 1, ncol = n + 1)  
  for (i in 0:n) {  
    Li <- c(1)  
    for (j in 0:n) {  
      if (i != j) {  
        Li <- c(0, Li) - c(Li * soporte[j+1], 0)  
        Li <- Li / (soporte[i+1] - soporte[j+1])  
      }  
    }  
    coefs[i+1,] <- Li  
  }  
  return(coefs)  
}
```

```
#Función para imprimir las bases de Lagrange
```

```
imprime_bases <- function(matriz_bases_Lagrange) {  
  for (j in 1:nrow(matriz_bases_Lagrange)) {  
    pol <- matriz_bases_Lagrange[j,]  
    cat(sprintf("L_%d = ", j-1))  
    for (i in 1:length(pol)) {  
      if (i > 1 && pol[i] > 0) cat(" + ")  
      cat(sprintf("%.4f", pol[i])) if (i > 1) cat(sprintf("x^%d", i-1))  
    }  
  }  
}
```

## Grupo M9

```
} cat("\n")  
}  
}
```

La siguiente función (`imprime_pol_canonico`) imprime un polinomio en forma canónica. Toma un vector de coeficientes y los muestra como un polinomio estándar, con los términos ordenados de menor a mayor grado. También creamos otra función (`evalua_pol_canonico`) que evalúa un polinomio en forma canónica para un conjunto dado de valores  $x$ . Utiliza una función interna para calcular el valor del polinomio en un punto y luego aplica esta función a todos los puntos de evaluación usando `sapply`.

#Función para imprimir el polinomio canónico

```
imprime_pol_canonico <- function(pol) {  
  for (i in 1:length(pol)) {  
    if (i > 1 && pol[i] >= 0) cat(" + ")  
    cat(sprintf("%.4f", pol[i]))  
    if (i > 1) cat(sprintf("x^%d", i-1))  
  }  
  cat("\n")  
}
```

#Función para evaluar el polinomio canónico

```
evalua_pol_canonico <- function(pol, x_eval) {  
  p <- function(x) {  
    resultado <- pol[1]  
    for (i in 2:length(pol)) {  
      resultado <- resultado + pol[i] * x^(i - 1)  
    }  
    return(resultado)  
  }  
  return(sapply(x_eval, p))  
}
```

Posteriormente utilizamos una función que construye el polinomio interpolador de Lagrange. Toma la matriz de coeficientes de las bases de Lagrange y los valores de la función en los puntos de soporte, y combina estos para formar el polinomio interpolador.

## Grupo M9

#Función para calcular el polinomio interpolador de Lagrange

```
pol_inter_lagrange <- function(matrizLs, fs) {  
  pol_interp <- numeric(ncol(matrizLs))  
  for (j in 1:nrow(matrizLs)) {  
    pol_interp <- pol_interp + matrizLs[j,] * fs[j]  
  }  
  return(pol_interp)  
}
```

Adicionalmente, podemos implementar una función que dibuja el gráfico del polinomio interpolador. Crea una función que representa el polinomio, genera una secuencia de puntos  $x$ , evalúa el polinomio en estos puntos, y luego dibuja la curva resultante junto con los puntos de soporte originales.

# Función para dibujar el polinomio canónico

```
dibujar_canonicos <- function(x_orig, coeficientes) {  
  polinomio <- function(x) {  
    suma <- 0  
    for (i in 1:length(coeficientes)) {  
      suma <- suma + coeficientes[i] * x^(i-1)  
    } return(suma)  
  }  
  x_min <- min(x_orig)  
  x_max <- max(x_orig)  
  rango <- x_max - x_min  
  x <- seq(x_min - 0.1rango, x_max + 0.1rango, length.out = 1000)  
  y <- sapply(x, polinomio)  
  plot(x, y, type = "l", col = "blue",  
       main = "Gráfica del Polinomio Canónico",  
       xlab = "x", ylab = "y",  
       xlim = c(min(x), max(x)), ylim = range(y))  
  points(x_orig, sapply(x_orig, polinomio), col = "red", pch = 19)  
}
```

## Grupo M9

Por último, hay que definir los datos de ejemplo.

```
# Datos de ejemplo (puedes modificar estos valores)
```

```
soporte <- c(-1, 0, 1, 2)
```

```
fs <- c(1, 0, 1, 4)
```

A continuación, se calculan los coeficientes de Lagrange, se imprimen las bases y el polinomio interpolador, se dibuja el gráfico, y finalmente se evalúa el polinomio en puntos específicos y se muestran los resultados.

```
# Puntos para evaluar el polinomio (puedes modificar estos valores)
```

```
x_eval <- seq(-2, 3, by = 5)
```

```
# Calcular los coeficientes de Lagrange
```

```
coefs <- lagrange_coefs(soporte)
```

```
# Imprimir las bases de Lagrange
```

```
cat("Bases de Lagrange:\n") imprime_bases(coefs)
```

```
# Calcular el polinomio interpolador
```

```
pol_interp <- pol_inter_lagrange(coefs, fs)
```

```
# Imprimir el polinomio interpolador en forma canónica
```

```
cat("\nPolinomio interpolador en forma canónica:\n") imprime_pol_canonico(pol_interp)
```

```
# Dibujar el polinomio interpolador
```

```
dibujar_canonicos(soporte, pol_interp)
```

```
# Evaluar el polinomio en los puntos especificados
```

```
y_eval <- evalua_pol_canonico(pol_interp, x_eval)
```

```
# Imprimir los resultados de la evaluación
```

```
cat("\nEvaluación del polinomio:\n")
```

```
for (i in 1:length(x_eval)) {
```

```
  cat(sprintf("f(%0.4f) = %0.4f\n", x_eval[i], y_eval[i]))
```

```
}
```

En este caso, el código sería introduciendo los datos directamente, sin pedirselos al usuario mediante la consola, pero también se puede hacer de esa forma:

```
# Parte interactiva que solicita datos al usuario
```

```
n <- as.integer(readline(prompt="Ingrese el número de puntos de soporte: "))
```

```
soporte <- numeric(n)
```

## Grupo M9

```
fs <- numeric(n)
cat("Ingrese los puntos de soporte y sus valores correspondientes:\n")
for(i in 1:n) {
  soporte[i] <- as.numeric(readline(prompt=sprintf("x[%d]: ", i)))
  fs[i] <- as.numeric(readline(prompt=sprintf("f(x[%d]): ", i)))
}
```

Para probarlo en R utilizar el documento nombrado como “sin datos”.