

Ejercicio 2

El proceso de fermentación en un biorreactor consta de tres fases:

1. **Fase de latencia:** en la que los microorganismos se adaptan al ambiente y el crecimiento es relativamente lento.
2. **Fase exponencial:** Los microorganismos comienzan a reproducirse más rápidamente debido a la disponibilidad de nutrientes.
3. **Fase estacionaria:** A medida que los nutrientes se agotan, la tasa de crecimiento disminuye y eventualmente se estabiliza.

Los científicos de un laboratorio han descrito este proceso mediante la siguiente función definida a tramos:

$$f(t) = \begin{cases} 0 & \text{si } t < 0 \\ R_{\text{lag}} t^2 & \text{si } 0 \leq t < T_{\text{lag}} & \text{Fase de latencia} \\ R_{\text{lag}} T_{\text{lag}}^2 + R_{\text{max}} (1 - e^{-k(t-T_{\text{lag}})}) & \text{si } T_{\text{lag}} \leq t < T_{\text{exp}} & \text{Fase exponencial} \\ f_{\text{exp}} e^{-\alpha(t-T_{\text{exp}})} & \text{si } t \geq T_{\text{exp}} & \text{Fase estacionaria} \end{cases}$$

Apartado a

(1 PTO)

Define en el script correspondiente al Ejercicio 2 las siguientes constantes (a modo de variables globales):

- $R_{\text{lag}} = 0.02$, siendo este parámetro la tasa de crecimiento en la fase de latencia, medido en (g/L/h³).
- $T_{\text{lag}} = 4$, duración en horas de la fase de latencia.
- $R_{\text{max}} = 1.2$, como la tasa máxima de crecimiento medida en (g/L/h)
- $k = 0.3$, como la constante de crecimiento de la fase exponencial, medida en h⁻¹
- $T_{\text{exp}} = 12$, como el momento en el que se produce la transición entre la fase exponencial y la estacionaria en h.
- $\alpha = 0.3$, como la tasa de decaimiento en h⁻¹
- $f_{\text{exp}} = R_{\text{lag}} T_{\text{lag}}^2 + R_{\text{max}} (1 - e^{-k(T_{\text{exp}} - T_{\text{lag}})})$, siendo el valor de la concentración de microorganismos en (g/L) en el momento de la transición entre la fase exponencial y la estacionaria.

```
tasa_crecimiento <- 0.02 # R_lag
```

```
fase_lag <- 4 # T_lag
```

```
tasa_max <- 1.2 # R_max
```

```
constante_crec <- 0.3 # k
```

```
tiempo_exp <- 12 # T_exp
```

```
tasa_deca <- 0.05 # alpha
```

Apartado b

(1.5 PTOS)

Define una función f que simule el comportamiento del proceso mediante el modelo que han propuesto los científicos. Esta función tiene como único parámetro de entrada el tiempo, t , y hace uso de todas variables globales definidas en el Apartado a.

```
modelo_fermentacion <- function(t) {  
  if (t >= 0 & t < fase_lag) {  
    return(tasa_crecimiento * t^2)  
  } else if (t >= fase_lag & t < tiempo_exp) {  
    return(tasa_crecimiento * fase_lag^2 + tasa_max * (1 - exp(-constante_crec * (t -  
fase_lag))))  
  } else if (t >= tiempo_exp) {  
    return(f_exp * exp(-tasa_deca * (t - tiempo_exp)))  
  } else {  
    return(0)  
  }  
}
```

Apartado c

(1.5 PTOS)

Comprueba la continuidad C^0 de la función $f(t)$. Para ello:

- Crea una secuencia de 201 elementos que vaya desde -1 hasta 20, para representar la variable temporal t .
- Evalúa $f(t)$ en la secuencia generada anteriormente y píntala en una gráfica (t en el eje de las abscisas y $f(t)$ en el de ordenadas).

```
tiempo <- seq(-1, 20, length.out = 201)
biomasa <- sapply(tiempo, modelo_fermentacion)
plot(
  tiempo, biomasa, type = "l", col = "green", lwd = 2,
  xlab = "Tiempo (h)", ylab = "Concentración de Biomasa (g/L)",
  main = "Crecimiento de Microorganismos durante Fermentación"
)
abline(v = c(fase_lag, tiempo_exp), col = "red", lty = 2)
text(0, max(biomasa) * 0.9, "Fase de Latencia", col = "red", pos = 4)
text(fase_lag, max(biomasa) * 0.9, "Fase Exponencial", col = "red", pos = 4)
text(tiempo_exp, max(biomasa) * 0.9, "Fase Estacionaria", col = "red", pos = 4)
```

Apartado d

(2 PTOS)

Estima la tasa de crecimiento de los microorganismos mediante $f'(t)$ en el instante $t^* = 10$ h. Para ello:

- Calcula el valor de la derivada de forma analítica, teniendo en cuenta que $f'(t) = R_{\max} \cdot k \cdot e^{-k(t-T_{\text{lag}})}$ durante la fase exponencial. Imprime el resultado de $f'(10)$ por pantalla.
- Calcula una estimación de la derivada mediante el método de los coeficientes indeterminados, empleando un soporte de 5 elementos equiespaciados en el intervalo $[T_{\text{lag}}, T_{\text{exp}}]$. Imprime el resultado por pantalla.

```
derivada_analitica <- function(t) {  
  return(tasa_max * constante_crec * exp(-constante_crec * (t - fase_lag)))  
}  
  
coeficientes_indeterminados <- function(s_points, x_0, orden_diff) {  
  abs_diff <- abs(s_points - x_0)  
  filtered_diff <- abs_diff[abs_diff != 0]  
  if (length(filtered_diff) == 0) {  
    stop()  
  }  
  h <- min(filtered_diff)  
  n_points <- length(s_points)  
  if (orden_diff >= n_points) {  
    stop(paste(  
      "El orden de la derivada ("  
      orden_diff,  
      ") debe ser menor que el número de puntos ("  
      n_points, ")."  
    ))  
  }  
}
```

```

thetas <- (s_points - x_0) / h
vandermonde_matrix <- matrix(0, nrow = n_points, ncol = n_points)
for (i in 1:n_points) {
  for (j in 1:n_points) {
    vandermonde_matrix[i, j] <- thetas[j]^(i - 1)
  }
}
diff_equal <- numeric(length = n_points)
diff_equal[orden_diff + 1] <- factorial(orden_diff)
coff_indet_diff <- solve(vandermonde_matrix, diff_equal * (1 / h^orden_diff))
return(coff_indet_diff)
}

puntos_soporte <- seq(fase_lag, tiempo_exp, length.out = 5)
val_derivada_10 <- derivada_analitica(10)
cat("Derivada analítica en t = 10:", val_derivada_10, "\n")
valores_soporte <- sapply(puntos_soporte, modelo_fermentacion)
coeficientes <- coeficientes_indeterminados(puntos_soporte, 10, 1)
derivada_estimada <- sum(coeficientes * valores_soporte)
cat("Derivada estimada en t = 10:", derivada_estimada, "\n")

```