Elementos de Cálculo Numérico

Primer Cuatrimestre 2026

Laboratorio N° 3: Convolución y FFT

Ejercicio 1 (Convolución circular.) Dados dos vectores $u, v \in \mathbb{C}^N$, su convolución circular es el vector $w \in \mathbb{C}^N$ con

$$w_k = \sum_{j=0}^{N-1} u_j \, v_{(k-j) \bmod N}, \quad k = 0, \ldots, N-1.$$

1. Programe una función `conv_circular_directa(u, v)` que calcule w usando la definición (dos bucles anidados, o un bucle y slicing). ¿Cuál es la complejidad de este algoritmo?

2. Programe una función `conv_circular_fft(u, v)` que calcule w usando la FFT:

   i. Calcular $\hat{u} = \mathtt{fft}(u)$ y $\hat{v} = \mathtt{fft}(v)$.
   ii. Multiplicar punto a punto: $\hat{w}_k = \hat{u}_k \cdot \hat{v}_k$.
   iii. Recuperar $w = \mathtt{ifft}(\hat{w})$.

   ¿Cuál es la complejidad de este algoritmo?

3. Genere vectores aleatorios $u, v \in \mathbb{R}^N$ para $N = 2^{10}$ y verifique que ambas funciones dan el mismo resultado (salvo errores de redondeo). ¿Cuál es el máximo de $|w_{\text{directa}} - w_{\text{fft}}|$?

4. Mida el tiempo de ejecución de ambas funciones para $N = 2^8, 2^{10}, 2^{12}, 2^{14}, 2^{16}$. Grafique los tiempos en escala log-log. ¿Se observa la diferencia $O(N^2)$ vs. $O(N \log N)$?

Ejercicio 2 (Convolución lineal via FFT.) Dados $u \in \mathbb{C}^M$ y $v \in \mathbb{C}^K$, su convolución lineal es el vector $w \in \mathbb{C}^{M+K-1}$ con

$$w_k = \sum_j u_j \, v_{k-j}, \quad k = 0, \ldots, M+K-2,$$

donde $u_j = 0$ para $j \notin \{0, \ldots, M-1\}$ y $v_j = 0$ para $j \notin \{0, \ldots, K-1\}$.

1. Programe una función `conv_lineal_directa(u, v)` que calcule w usando la definición.

2. Para calcular la convolución lineal usando FFT, hay que realizar *zero-padding*: extender ambos vectores a longitud $N \geq M + K - 1$ completando con ceros.

   i. Explique por qué si $N < M+K-1$, la convolución circular de los vectores extendidos **no** coincide con la convolución lineal (se produce *aliasing temporal*).

ii. Programe una función `conv_lineal_fft(u, v)` que haga el zero-padding adecuado, aplique la convolución circular via FFT, y devuelva los primeros $M+K-1$ elementos.

3. Genere $u \in \mathbb{R}^{100}$ y $v \in \mathbb{R}^{50}$ aleatorios. Verifique que `conv_lineal_fft` da el mismo resultado que `conv_lineal_directa`. Compare también con `numpy.convolve(u, v)`.

4. Ilustre el efecto del aliasing: calcule la convolución circular (sin zero-padding) de u y v extendidos a $N = \max(M, K)$ y compare con la convolución lineal correcta. Grafique ambos resultados superpuestos. ¿Dónde se concentran las diferencias?

5. Mida el tiempo de ejecución de ambos métodos para $M = K = 2^8, 2^{10}, 2^{12}, 2^{14}$. ¿A partir de qué tamaño la versión FFT es más rápida?