Cálculo de sumas trigonométricas por medio de la transformada finita de Fourier

Objetivos. Mostrar que las sumas trigonométricas de la forma

$$\sum_{k=-n+1}^{n-1} a_k \, \mathrm{e}^{k\,\mathrm{i}\,\vartheta},$$

en los puntos de la forma $\frac{2\pi}{n}$, se pueden expresar en términos de la transformada finita de Fourier.

Ejercicio 1. Supongamos que los vectores $b, c \in \mathbb{C}^n$ contienen los coeficientes $a_0, \ldots, a_{n-1}$ y $a_0, \ldots, a_{-n+1}$, respectivamente. Escribir una función que calcule los valores de la suma

$$\sum_{k=-n+1}^{n+1} a_k \, \mathrm{e}^{\mathrm{i}\,k\vartheta}$$

en cada uno de los puntos $2\pi j/n$ con $j = 0, \ldots, n-1$.

```
function [v] = trig_sum_values(b, c),
   n = length(b);
   for j = 1 : n,
      ind = (0 : n - 1)';
      powers = exp(2 * pi * i * ind);
      possum = (powers.') * b;
      negsum = powers' * c;
      v(j) = possum + negsum - b(1);
   end
end
```

Proposición 2. *Sea $n \in \mathbb{N}$ y sean $a_{-n+1}, \ldots, a_{n-1} \in \mathbb{C}$. Consideremos la función*

$$f(\vartheta) := \sum_{k=-n+1}^{n-1} a_k \, \mathrm{e}^{k\,\mathrm{i}\,\vartheta}.$$

Denotemos por x_j al punto $\frac{2\pi j}{n}$. Denotemos por b y c los vectores

$$b = \big[a_k\big]_{k=0}^{n-1}, \quad c = \big[a_{-k}\big]_{k=0}^{n-1}.$$

Entonces para cada j en $\{0, \ldots, n-1\}$

$$f(x_j) = \overline{(F_n \bar{b})_j} + (F_n c)_j - b_0. \tag{1}$$

Demostración. Notamos que

$$f(x_j) = \sum_{k=0}^{n-1} a_k \, \mathrm{e}^{k\,\mathrm{i}\,\frac{2\pi j}{n}} + \sum_{k=0}^{n-1} a_{-k} \, \mathrm{e}^{-k\,\mathrm{i}\,\frac{2\pi j}{n}} - a_0.$$

Transformamos la primera suma:

$$\sum_{k=0}^{n-1} a_k \, \mathrm{e}^{k\,\mathrm{i}\,\frac{2\pi j}{n}} = \sum_{k=0}^{n-1} a_k \overline{\omega_n^{jk}} = \overline{(F_n \overline{b})_j}.$$

Transformamos la segunda suma:

$$\sum_{k=0}^{n-1} a_{-k} \, \mathrm{e}^{-k\,\mathrm{i}\,\frac{2\pi j}{n}} = (F_n c)_j.$$

□

Ejercicio 3. En algún lenguaje de programación escribir una función que a partir de los vectores b y c calcule los valores de f en los puntos $2\pi j/n$. Idea de solución.

```
function [v] = trig_sum_values_via_fft(b, c),
   prodpos = conj(fft(conj(b)));
   prodneg = ???;
   constantrep = b(1) * ones(size(b));
   v = ???;
end
```

Escribir una función que haga la comprobación.

Ejercicio 4. En algún lenguaje de programación escribir una función que haga una comprobación de la función `trig_sum_values_via_fft` comparando su resultado con el resultado de la función `trig_sum_values`. Aplicar estas funciones a vectores aleatorios `b` y `c`, cuyas primeras componentes deben coincidir. Medir el tiempo de ejecución para `n` grande.

Ejercicio 5. En algún lenguaje de programación escribir una función que haga una comprobación de la función `trig_sum_values_via_fft` aplicándola a los coeficientes de Fourier del núcleo de Poisson y comparando el resultado con el valor verdadero del núcleo de Poisson. Calcular el error de la aproximación. También se recomienda calcular este error de manera teórica.