Ejercicio en clases
Scientific Computing Team - Semana - 09

1. El Teorema de interpolación de la transformada discreta de coseno indica lo siguiente:

Thm 1. *Sea* $\mathbf{x} = [x_0, x_1, \dots, x_{n-1}]^T$ *un vector con* n *coeficientes reales. Se define* $\mathbf{y} = [y_0, y_1, \dots, y_{n-1}]^T = C\,\mathbf{x}$*, donde* C *corresponde a la matriz asociada a la Transformada Discreta de Coseno (DCT del inglés Discrete Cosine Transform) de orden* n*. Entonces la función:*

$$P_n(t) = \frac{1}{\sqrt{n}}\,y_0 + \sqrt{\frac{2}{n}}\sum_{k=1}^{n-1} y_k\,\cos\left(\frac{k\,(2\,t+1)\,\pi}{2\,n}\right),$$

satisface que $P_n(j) = x_j$ *para todo* $j \in \{0, 1, \dots n-1\}$ *y los coeficientes de la matriz* $C \in \mathbb{R}^{n\times n}$ *se define de la siguiente forma:*

$$C_{i,j} = \sqrt{\frac{2}{n}}\,a_i\,\cos\left(\frac{i\,(2\,j+1)\,\pi}{2\,n}\right),$$

para $i \in \{0, 1, \dots n-1\}$ *y* $j \in \{0, 1, \dots n-1\}$*, donde*

$$a_i = \begin{cases} \dfrac{1}{\sqrt{2}}, & si\ i = 0, \\ 1, & si\ i \in \{1, 2, \dots, n-1\}. \end{cases}$$

Por simplicidad se incluye de forma explícita la matriz C,

$$C = \sqrt{\frac{2}{n}}\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \dots & \frac{1}{\sqrt{2}} \\ \cos\left(\frac{\pi}{2\,n}\right) & \cos\left(\frac{3\,\pi}{2\,n}\right) & \dots & \cos\left(\frac{(2\,n-1)\,\pi}{2\,n}\right) \\ \cos\left(\frac{2\,\pi}{2\,n}\right) & \cos\left(\frac{6\,\pi}{2\,n}\right) & \dots & \cos\left(\frac{2\,(2\,n-1)\,\pi}{2\,n}\right) \\ \vdots & \vdots & \vdots & \vdots \\ \cos\left(\frac{(n-1)\,\pi}{2\,n}\right) & \cos\left(\frac{(n-1)\,3\,\pi}{2\,n}\right) & \dots & \cos\left(\frac{(n-1)\,(2\,n-1)\,\pi}{2\,n}\right) \end{bmatrix}$$

En resumen, podemos concluir que necesitamos evaluar recurrentemente la función $f(x) = \cos(\pi\,x)$ al utilizar la DCT. Más aún, es posible notar que para construir la transformada discreta de coseno para $n = 100$ necesitamos evaluar el coeficiente $C_{n-1,n-1} = \cos\left(\frac{(n-1)\,(2\,n-1)\,\pi}{2\,n}\right)$, lo que es $\cos\left(\frac{19701}{200}\,\pi\right) \approx \cos(309.462584341862582954) \approx -0.015707317311820$. Entonces, si consideramos que trabajaremos con vectores de dimensión menor o igual a 100, significa que necesitamos evaluar la función $\cos(\cdot)$ para valores en el intervalo $[0, 100\,\pi]$.

Adicionalmente sabemos que,

$$\left|\frac{\mathrm{d}^m f(x)}{\mathrm{d}x^m}\right| = \begin{cases} \pi^m\,|\sin(\pi\,x)|, & \text{si } m \text{ es impar}, \\ \pi^m\,|\cos(\pi\,x)|, & \text{si } m \text{ es par}. \end{cases}$$

Preguntas:

(a) Proponga un algoritmo basado en interpolación polinomial que permita evaluar la función $f(x)$ para $x \in [0, 100]$ que asegure lo siguientes requerimientos:

Req. 1 Un error de interpolación menor o igual a $\varepsilon > 0$. Usted debe explicar que se debe hacer para asegurar esto y cómo planea determinar lo que necesite para cumplir con lo solicitado.

Req. 2 Que el interpolador construido tenga una cota superior de error lo más pequeña posible.

Su respuesta debe:

- Explicar claramente como **construirá** su interpolador polinomial.
- Explicar claramente como se debe **utilizar** su interpolador polinomial.

Recuerde que cada algoritmo tiene distintos requerimientos, por lo que verifique que usted efectivamente describe explícitamente cada componente del algoritmo elegido para que funcione completamente. En caso de no cumplir con alguno de los 2 requerimientos completamente, se entregarán puntos parciales.

(b) Explique cómo puede utilizar la interpolación polinomial anterior para evaluar $\sin(\pi\,x)$ sin construir un nuevo interpolador.

- Hint 1: It is important to recall that the cosine function is periodic, so, what can you do if you want to evaluate it for values greater that $2\,\pi$? The same applies for negative values, but they are not needed for this problem.
- Hint 2: You may find useful to consider the modulus operator, which returns the reminder of the operation. For instance in NumPy we can performe the following computation `np.mod(6.2,2.5)` which returns `1.2`.
- Hint 3: The lowest upper bound should contain the coefficient π^n but not $\pi^{2\,n}$ nor other additional coefficients that makes it larger. In this case n corresponds to the number of points used in the interpolation.