This is similar to the simple example given in class. The goal is to develop fast algorithms for the following matrix-vector product

\[ v = Au, \] (1)

where

\[ v_i = u_i + \sum_{j=1}^{N} u_j \cos^n (x_i - x_j), \quad j = 1, \ldots, N, \quad \{x_i\} \in [0, 2\pi) \] (2)

and

\[
\begin{pmatrix}
1 & \cos^n(x_1 - x_2) & \ldots & \cos^n(x_1 - x_N) \\
\cos^n(x_2 - x_1) & 1 & \ldots & \cos^n(x_2 - x_N) \\
& \ldots & \ldots & \ldots \\
\cos^n(x_N - x_1) & \cos^n(x_N - x_2) & \ldots & 1
\end{pmatrix}, \quad u = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix}, \quad v = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix}. \] (3)

Here \( x_1, \ldots, x_N, u_1, \ldots, u_N \), are given. The matrix dimension \( N > 0 \), and the power, \( n > 0 \), are given (fixed) positive integers.

1. Derive a formula that can be used for developing a “fast” \( O(N) \) method for arbitrary \( n \).

2. Write a program that implements both straightforward computation based on Eq. (2) and the “Fast” method. Use Matlab. Be sure that you can vary \( x_1, \ldots, x_N, \) and \( u_1, \ldots, u_N \).

3. Check that both methods produce the same results (within machine precision). Plot the absolute maximum error between the straightforward and the “Fast” method for \( n = 5 \) and \( N \) varying between \( 10^2 \) and \( 10^3 \).

4. Plot a comparison of the CPU time required by the straightforward and the “Fast” methods for \( n = 5 \) and \( N \) varying between \( 10^2 \) and \( 10^3 \) for the straightforward method, and with \( N \) varying between \( 10^2 \) and \( 10^4 \) for the “Fast” method. Make the plots log-log. Also plot lines corresponding to linear and quadratic dependences of the CPU time on \( N \) and compare with your computational results.

5. Make a conclusion about the asymptotic complexity of each method.

Hints

1. Use the trigonometric identity

\[ \cos (x_i - x_j) = \cos x_i \cos x_j + \sin x_i \sin x_j. \] (4)

2. Combine with the Newton binomial theorem

\[ (a + b)^n = a^n + \binom{n}{1} a^{n-1} b + \binom{n}{2} a^{n-2} b^2 + \ldots + b^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k, \] (5)

where the binomial coefficients are

\[ \binom{n}{k} = \frac{n(n-1) \ldots (n-k+1)}{1 \cdot 2 \cdot \ldots \cdot k} = \frac{n!}{(n-k)!k!}. \] (6)

3. For CPU time measurement use the Matlab function \texttt{cputime} (see Matlab help).