

Compute the matrix-vector product

$$\mathbf{v} = \Phi \mathbf{u}, \quad v_j = \sum_{i=1}^N \Phi_{ji} u_i, \quad j = 1, \dots, M, \quad (1)$$

with absolute error $\epsilon < 10^{-6}$, where

$$\Phi = \begin{pmatrix} \Phi_{11} & \Phi_{12} & \dots & \Phi_{1N} \\ \Phi_{21} & \Phi_{22} & \dots & \Phi_{2N} \\ \dots & \dots & \dots & \dots \\ \Phi_{M1} & \Phi_{M2} & \dots & \Phi_{MN} \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \dots \\ u_N \end{pmatrix}, \quad \mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \dots \\ v_M \end{pmatrix}, \quad (2)$$

$$\Phi_{ji} = \frac{1}{y_j - x_i}, \quad i = 1, \dots, N, \quad j = 1, \dots, M. \quad (3)$$

and x_1, \dots, x_N and u_1, \dots, u_N are uniformly distributed on $[0,1]$, $M = N - 1$, and each y_j is located between the closest x_i 's on each side, $j = 1, \dots, N - 1$ using the Single Level FMM algorithm.

1. Draw a rough sketch of the SL-FMM algorithm.
2. Use the routines developed in Homework 4.
3. Evaluate the truncation number, $p(K, N)$, that provides the specified accuracy as a function of the number of boxes K and of N .
4. Write a program that implements both straightforward multiplication based on Eq. (1) and SL-FMM.
5. Compare the theoretical optimal number of boxes with the actual one achieved by your program.
6. Provide a dependence of the CPU time required by the Pre-FMM as a function of K for $N = 5000$ ($10 < K < 200$). Determine K_{opt} experimentally and compare with the theoretical evaluations (use the actual p). Scale $K_{opt}(N)$ for computations with varying N . Plot your scaled function $K_{opt}(N)$.
7. Provide a graph of actual error (between the standard and the fast method with $K = K_{opt}(N)$) for N varying between 10^2 and 5×10^3 and the truncation number used.
8. Provide a graph that compares the CPU time required by the straightforward method and the Pre-FMM for N varying between 10^2 and 10^3 for straightforward and N varying between 10^2 and 10^4 for the optimized SL-FMM. Compare results with theoretical complexities of the algorithms.