Homework 3, AMSC698R/CMSC878R/MAIT622

Due September 26, 2006

1 Pre-FMM using Local Expansions

Compute the matrix-vector product
\[ \mathbf{v} = \Phi \mathbf{u}, \quad v_j = \sum_{i=1}^{N} \Phi_{ji} u_i, \quad j = 1, ..., M, \] (1)

with absolute error \( \epsilon < 10^{-6} \), where
\[ \Phi = \begin{pmatrix} \Phi_{11} & \Phi_{12} & \ldots & \Phi_{1N} \\ \Phi_{21} & \Phi_{22} & \ldots & \Phi_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_{M1} & \Phi_{M2} & \ldots & \Phi_{MN} \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix}, \quad \mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_M \end{pmatrix}, \] (2)

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

and \( x_1, ..., x_N \) and \( u_1, ..., u_N \) are uniformly distributed on \([0,1]\), \( M = N - 1 \), and each \( y_j \) is located between the closest \( x_i \)'s on each side, \( j = 1, ..., N - 1 \) using the Pre-FMM that employs \( R \)-expansions near the centers of the target boxes.

1. Draw a rough sketch of the Pre-FMM algorithm.

2. Evaluate the truncation number, \( p(K, N) \), that provides the specified accuracy as a function of the number of boxes \( K \) and of \( N \).

3. Evaluate theoretically the optimal number of boxes \( K_{opt}(N) \) for space division based on the obtained evaluations of \( p \) for specified accuracy.

4. Write a program which provides the \( R \)-expansion coefficients for a given target box (or target box center) and a source in arbitrary position in the domain. Test its accuracy

5. Write a program that implements both straightforward multiplication based on Eq. (1) and Pre-FMM that uses \( R \)-expansions.

6. Provide a graph of the absolute maximum error between the two programs for \( N = 10^3 \), \( K \) varying between 10 and 100, and \( p \) from your theoretical evaluations. Compare the accuracy results with theory. You may find that the theoretical \( p \) may be much larger than the one needed in practice. In this case you may (or may not) reduce \( p \) and use some experimental values to proceed further.

7. Provide a dependence of the CPU time required by the Pre-FMM as a function of \( K \) for \( N = 10^3 \) (10 < \( K < 100 \)). 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 \( K_{opt}(N) \).

8. 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 10\(^3\) and the truncation number used.

9. 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 Pre-FMM. Compare results with theoretical complexities of the algorithms.