

**FMM-YUKAWA: AN ADAPTIVE FAST MULTIPOLE METHOD
FOR SCREENED COULOMB INTERACTIONS**

By

Jingfang Huang

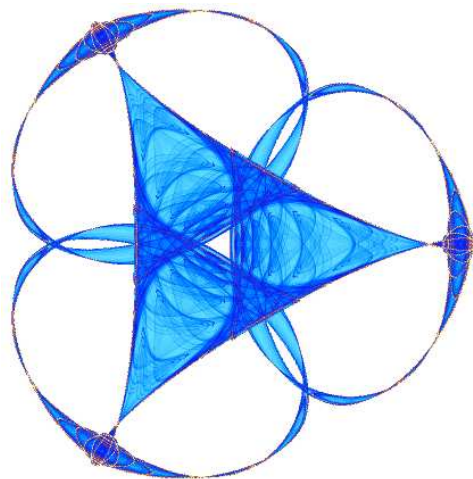
Jun Jia

and

Bo Zhang

IMA Preprint Series # 2258

(June 2009)



INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS

UNIVERSITY OF MINNESOTA
400 Lind Hall
207 Church Street S.E.
Minneapolis, Minnesota 55455-0436

Phone: 612/624-6066 Fax: 612/626-7370

URL: <http://www.ima.umn.edu>

FMM-Yukawa: An Adaptive Fast Multipole Method for Screened Coulomb Interactions

Jingfang Huang^a, Jun Jia^b, Bo Zhang^{a,1}

^a*CB# 3250, Phillips Hall, Department of Mathematics, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3250*

^b*Oak Ridge Nation Lab, Oak Ridge, TN*

Abstract

A Fortran program package is introduced for the rapid evaluation of the screened Coulomb interactions of N particles in three dimensions. The method utilizes an adaptive oct-tree structure, and is based on the new version of fast multipole method in which the exponential expansions are used to diagonalize the multipole-to-local translations. The program and its full description, as well as several closely related packages are also available at <http://www.fastmultipole.org/>. This paper is a brief review of the program and its performance.

PACS: 02.30.Em; 02.30.Rz; 02.60.Dc; 24.10.Cn.

Key words: Fast Multipole Method; Screened Coulomb Potential; Yukawa Potential; Diagonal Translation; Exponential Sums.

PROGRAM SUMMARY

Manuscript Title: FMM-Yukawa: An Adaptive Fast Multipole Method for Screened Coulomb Interactions.

Authors: Jingfang Huang, Jun Jia, Bo Zhang

Program Title: FMM-Yukawa

Journal Reference:

Catalogue identifier:

Licensing provisions: GPL 2.0.

Programming language: Fortran77 and Fortran90.

Computer: Any.

Operating system: Any.

RAM: Depends on the number of particles, their distribution, and the adaptive tree structure.

Supplementary material: Check <http://www.fastmultipole.org/>

¹ Corresponding author, Email: zhangb@email.unc.edu, Tel: 919-962-8495

Keywords: Fast Multipole Method; Screened Coulomb Potential; Yukawa Potential; Diagonal Translation; Exponential Sums.

PACS: 02.30.Em; 02.30.Rz; 02.60.Dc; 24.10.Cn.

Classification: 4.8 Linear Equation and Matrices; 4.12 Other Numerical Methods.

Nature of problem: To evaluate the screened Coulomb potential and force field of N charged particles, and to evaluate a convolution type integral where the Green's function is the fundamental solution of the modified Helmholtz equation.

Solution method: An adaptive oct-tree is generated, and a new version of fast multipole method is applied in which the "multipole-to-local" translation operator is diagonalized.

Restrictions: Only three and six significant digits accuracy options are provided in this version.

Unusual features: Most of the codes are written in Fortran77. Functions for memory allocation from Fortran90 and above are used in one subroutine.

Running time: The running time varies depending on the number of particles (denoted by N) in the system and their distribution. The running time scales linearly as a function of N for nearly uniform particle distributions. For three digits accuracy, the solver breaks even with direct summation method at about $N = 750$.

References: L. Greengard and J. Huang, "A New Version of the Fast Multipole Method for Screened Coulomb Interactions in Three Dimensions", J. Comput. Phys. 180, 642-658 (2002).

1 Introduction

In this paper, we introduce a software package for calculating the long range screened Coulomb interactions of N particles using the new version of fast multipole method (FMM). Given N charged particles each carrying a charge q_i located at $\mathbf{x}_i = (x_i, y_i, z_i)$ and $\lambda \in \mathbb{R}^+$, we want to efficiently evaluate the potential field

$$\Phi(\mathbf{x}_j) = \sum_{\substack{i=1 \\ i \neq j}}^N q_i \cdot \frac{e^{-\lambda \|\mathbf{x}_j - \mathbf{x}_i\|}}{\|\mathbf{x}_j - \mathbf{x}_i\|} \quad (1)$$

and the corresponding force field $\mathbf{F} = \nabla\Phi$. This calculation is required in various problems in physics, chemistry, and biology, when Coulomb forces are damped by screening effects. In nuclear physics, the Green's function is referred to as the Yukawa potential, and the corresponding partial differential equation $\nabla^2\Phi - \lambda^2\Phi = f(\mathbf{x})$ occurs in implicit marching schemes for the Navier-Stokes equations and parabolic equations, in Debye-Huckel theory, and is referred to as the linearized Poisson-Boltzmann equation in biophysics and biochemistry [9,11,15].

Notice that direct calculation of the N -body interactions in Eq. (1) requires prohibitive $O(N^2)$ operations. In the last two decades, however, several novel fast summation algorithms have been developed to reduce the cost to $O(N \log N)$ or the asymptotically optimal $O(N)$. These algorithms include the hierarchical "tree code" [1,2], fast Fourier transform (FFT) based algorithms such as the precorrected FFT (pFFT)[14] and the particle-mesh Ewald (PME) methods[5], the hierarchical SVD method[10], and FFT on multipoles[12,13]. Further improvements show that asymptotically optimal $O(N)$ complexity can be achieved by using the wavelet techniques or the fast multipole method (FMM) [6]. However, as revealed by previous numerical experiments, although asymptotically optimal, the original FMM [6] turns out to be less efficient for problem sizes of current interest when compared with the tree code and FFT based $O(N \log N)$ techniques, due to the huge prefactor in $O(N)$. In 1997, Greengard and Rokhlin introduced a new version of FMM [7] for the Laplace equation to further accelerate the performance of the FMM. Compared with the original FMM, the plane wave expansion based diagonal translation operators dramatically reduce the prefactor in the $O(N)$ new version FMM, especially in three dimensions where a break-even point of approximately $N = 750$ for three digits precision is numerically observed. In this paper, we generalize the new version of fast multipole method to Eq. (1) and introduce the software package

FMM-Yukawa. Perhaps due to the complexity in theory and programming, we are unaware of any previous open source implementations of the new version FMM for the Yukawa potential.

This paper is organized as follows. In Section 2, we summarize the classical theory of multipole, local, and plane wave expansions for screened Coulomb potentials, and introduce the translation operators. In Section 3, we present the pseudo-code to explain the overall algorithm structure, and provide installation instructions and a sample driver file. In Section 4, we illustrate the performance of the FMM-Yukawa solver with several numerical examples.

2 Theoretical Background

Assuming all particles are located inside the unit box centered at the origin, which is referred to as refinement level 0, we can build an oct-tree structure consisting of a hierarchy of boxes. In the structure, refinement level $l + 1$ is obtained from level l by subdividing each box at level l (which contains more than a prescribed number s of particles) into eight cubic boxes of equal size. We remove the empty child boxes and recursively refine the oct-tree structure until each childless box contains no more than s particles. When the particles are nearly uniformly distributed, the oct-tree has roughly $\log N$ levels for N source particles. In the oct-tree structure, a box c is said to be a *child* of box b , if box c is obtained by a single subdivision of box b . Box b is said to be the *parent* of box c .

For particles inside each box in the oct-tree structure, a multipole expansion can be constructed to account for its contribution to “far-field” boxes. Assume box c centered at the origin contains J particles of strengths q_1, q_2, \dots, q_J , then the potential at a point $\mathbf{x} = (r, \theta, \phi)$ outside box c is given by a multipole expansion

$$\begin{aligned} \Phi(\mathbf{x}) &= \sum_{i=1}^J q_i \cdot \frac{e^{-\lambda \|\mathbf{x} - \mathbf{x}_i\|}}{\|\mathbf{x} - \mathbf{x}_i\|} = \frac{2\lambda}{\pi} \sum_{i=1}^J q_i \cdot k_0(\lambda \|\mathbf{x} - \mathbf{x}_i\|) \\ &= \sum_{n=0}^{\infty} \sum_{m=-n}^n M_n^m k_n(\lambda r) \cdot Y_n^m(\theta, \phi), \end{aligned} \quad (2)$$

where the multipole coefficients are

$$M_n^m = 8\lambda \sum_{i=1}^J q_i \cdot i_n(\lambda \rho_i) \cdot Y_n^{-m}(\alpha_i, \beta_i). \quad (3)$$

In the formulas, Y_n^m is the spherical harmonics of degree n and order m defined

by

$$Y_n^m(\theta, \phi) = \sqrt{\frac{2n+1}{4\pi}} \sqrt{\frac{(n-|m|)!}{(n+|m|)!}} \cdot P_n^{|m|}(\cos \theta) e^{im\phi}, \quad (4)$$

P_n^m is the associated Legendre function defined using the usual Legendre polynomial $P_n(x)$ by the Rodrigues' formula

$$P_n^m(x) = (-1)^m (1-x^2)^{m/2} \frac{d^m}{dx^m} P_n(x),$$

and the modified spherical Bessel and modified spherical Hankel functions $i_n(r), k_n(r)$ are defined in terms of the usual Bessel function $J_\nu(z)$ via $I_\nu(r) = i^{-\nu} J_\nu(ir)$ ($i = \sqrt{-1}$), $K_\nu(r) = \frac{\pi}{2 \sin \nu \pi} [I_{-\nu}(r) - I_\nu(r)]$, $i_n(r) = \sqrt{\frac{\pi}{2r}} I_{n+1/2}(r)$, and $k_n(r) = \sqrt{\frac{\pi}{2r}} K_{n+1/2}(r)$. In particular, $k_0(\lambda r) = \frac{\pi}{2} \frac{e^{-\lambda r}}{\lambda r}$ is the Green's function for the modified Helmholtz equation (up to a constant). In the numerical simulations, the expansion in Eq. (2) must be truncated. For a prescribed accuracy requirement, the number of terms in the expansion usually depends on the ratio of \mathbf{x} to the box size. When \mathbf{x} locates in boxes at the same level of c which do not share a boundary point with c (the so-called "well-separated" or "far-field" boxes of c in the oct-tree structure), 21 terms in the first summation usually guarantees 7 digits accuracy, while 42 terms for double precision accuracy.

The multipole expansion of box c collects contributions from particles in c to the far-field boxes. Alternatively, suppose J sources of strengths q_1, q_2, \dots, q_J are located at the points $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_J$ in far-field boxes of c , with spherical coordinates $(\rho_1, \alpha_1, \beta_1), (\rho_2, \alpha_2, \beta_2), \dots, (\rho_J, \alpha_J, \beta_J)$ respectively. Then, for any point \mathbf{x} in box c with coordinates (r, θ, ϕ) , the potential $\Phi(\mathbf{x})$ generated by the "far-field" sources q_1, q_2, \dots, q_J can be represented as a local expansion of c as in

$$\Phi(\mathbf{x}) = \sum_{j=0}^{\infty} \sum_{k=-j}^j L_j^k i_j(\lambda r) \cdot Y_j^k(\theta, \phi), \quad (5)$$

where

$$L_j^k = 8\lambda \sum_{l=1}^N q_l k_j(\lambda \rho_l) \cdot Y_j^{-k}(\alpha_l, \beta_l). \quad (6)$$

Furthermore, for a prescribed accuracy requirement, the number of terms in the truncated local expansion is similar to that in the multipole expansions.

In the original fast multipole method [6], a divide-and-conquer strategy is applied to form all the multipole and local expansions for the boxes in the oct-tree structure. First, an upward sweep (from finest level to refinement level 0) is performed to form all the multipole expansions. For childless boxes, Eqs. (2-3) are applied to directly form the multipole expansion about the box center using the particles inside the box. For each parent box, instead of communicating with the particles, the multipole expansion is derived by shifting and merging its children’s multipole expansions, using the following “multipole-to-multipole” (\mathcal{T}_{MM}) translation operator:

Definition 1 (\mathcal{T}_{MM} Translation) *Consider a box b centered at the origin and its child c centered at $\mathbf{x}_0 = (\rho, \alpha, \beta)$, and a point $\mathbf{x} = (r, \theta, \phi)$ in the far-field boxes of b . Assume the potential due to charges in box c is given by the multipole expansion*

$$\Phi(\mathbf{x}) = \sum_{n=0}^{\infty} \sum_{m=-n}^n O_n^m k_n(\lambda r') \cdot Y_n^m(\theta', \phi'), \quad (7)$$

where (r', θ', ϕ') are the spherical coordinates of the vector $\mathbf{x} - \mathbf{x}_0$. Then, the field can also be described by a shifted multipole expansion:

$$\Phi(\mathbf{x}) = \sum_{n=0}^{\infty} \sum_{m=-n}^n M_n^m k_n(\lambda r) \cdot Y_n^m(\theta, \phi). \quad (8)$$

The linear operator mapping the old multipole coefficients $\{O_n^m\}$ to the new multipole coefficients $\{M_n^m\}$ is denoted by \mathcal{T}_{MM} .

After forming all the multipole expansions in the upward sweep, a downward sweep (from refinement level 0 to finest level) is performed to form the local expansion for each box which contains its far-field particle contributions. For a box c at level l , a translation is first performed to shift its parent b ’s local expansion (which contains particles information from b ’s far-field) to an equivalent local expansion centered at box c ’s center, using the “local-to-local” (\mathcal{T}_{LL}) operator as follows:

Definition 2 (\mathcal{T}_{LL} Translation) *Consider a point $\mathbf{x} = (r, \theta, \phi)$ in box c and the local expansion of c ’s parent box b*

$$\Phi(\mathbf{x}) = \sum_{n=0}^p \sum_{m=-n}^n L_n^m i_n(\lambda r) \cdot Y_n^m(\theta, \phi). \quad (9)$$

Assume box c is centered at $\mathbf{x}_0 = (\rho, \alpha, \beta)$, then the local expansion of c is

given by

$$\Phi(\mathbf{x}) = \sum_{n=0}^p \sum_{m=-n}^n N_n^m i_n(\lambda r') \cdot Y_n^m(\theta', \phi'), \quad (10)$$

where (r', θ', ϕ') are the spherical coordinates of the vector $\mathbf{x} - \mathbf{x}_0$. The linear operator mapping the old local coefficients $\{L_n^m\}$ to the new local coefficients $\{N_n^m\}$ is denoted by \mathcal{T}_{LL} .

In the second step of the downward pass, box c receives contributions from particles which are located in a region formed by subtracting its parent b 's far-field from its far-field. Box in this region is usually referred to as the “interaction list” box. Instead of directly communicating with particles in the interaction list, a “multipole-to-local” translation is performed to convert the multipole expansions of interaction list boxes directly to a local expansion which will be merged to c 's local expansion. The \mathcal{T}_{ML} translation operator is defined as follows:

Definition 3 (\mathcal{T}_{ML} Translation) Suppose box d centered at $X_0 = (\rho, \alpha, \beta)$ is an interaction list box of c , and its multipole expansion is given by the same formula as in Eq. (7). Then for any point $\mathbf{x} = (r, \theta, \phi)$ in c centered at the origin, the potential due to charges in b can be described by a local expansion

$$\Phi(\mathbf{x}) = \sum_{j=0}^{\infty} \sum_{k=-j}^j L_j^k i_j(\lambda r) \cdot Y_j^k(\theta, \phi). \quad (11)$$

The linear operator mapping the multipole coefficients $\{M_n^m\}$ to the local coefficients $\{L_n^m\}$ is denoted by \mathcal{T}_{ML} .

After the downward sweep, we evaluate the local expansion for particles in each childless box, and compute the “near-field” (neighbor) interactions directly. Notice that the particles are only used at the finest level, and interactions are accounted for through the $O(N)$ boxes at higher levels, the fast multipole method is therefore asymptotically optimal $O(N)$ in operations. However, as the “multipole-to-local” translation requires prohibitive $189p^4$ operations for each box (189 is the number of boxes in its interaction list, and we assume p^2 terms are used in each expansion), the huge prefactor makes the original FMM less competitive with the tree code and other FFT based methods. Instead, similar to the new version of FMM introduced by Greengard and Rokhlin [7] for the Laplace equation. We introduce the plane wave expansions in the FMM-Yukawa solver to “diagonalize” the \mathcal{T}_{ML} operator. In three dimensional space, six different exponential expansions are introduced for the six directions of a box c . For the interaction boxes in the “upward” direction (particles in the region defined by $1 \leq z \leq 4$, $0 \leq \sqrt{x^2 + y^2} \leq 4\sqrt{2}$, assuming the size of

box c is rescaled to 1), the multipole expansion of c can be translated to an exponential expansion by the “multipole-to-exponential” (\mathcal{T}_{ME}) translation operator as follows (other directions are similar and we omit the details):

Definition 4 (\mathcal{T}_{ME} Translation) For a point X in the “upward” direction with spherical coordinates (r, θ, ϕ) , assume the potential $\Phi(X)$ is approximated by the truncated multipole expansion (due to charges in box c centered at the origin) with error bound ε

$$\left| \Phi(X) - \sum_{n=0}^p \sum_{m=-n}^n O_n^m \cdot Y_n^m(\theta, \phi) k_n(\lambda r) \right| < \varepsilon.$$

Then with error bound $O(\varepsilon)$, $\Phi(X)$ can be approximated by

$$\lambda \sum_{k=1}^{s(\varepsilon)} \sum_{j=1}^{M_k} W(k, j) \cdot e^{-(u_k + \lambda) \cdot z} \cdot e^{i\sqrt{u_k^2 + 2u_k \lambda} \cdot (x \cdot \cos(\alpha_{j,k}) + y \cdot \sin(\alpha_{j,k}))} \quad (12)$$

where (x, y, z) are the Cartesian coordinates of X , the coefficients $W(k, j)$ is given by

$$\frac{\pi w_k}{2\lambda M_k} \sum_{m=-p}^p i^{|m|} \cdot e^{im \cdot \alpha_{j,k}} \sum_{n=|m|}^p O_n^m \sqrt{\frac{2n+1}{4\pi}} \sqrt{\frac{(n-|m|)!}{(n+|m|)!}} P_n^{|m|} \left(\frac{\lambda + u_k}{\lambda} \right),$$

for $k = 1, \dots, s(\varepsilon)$, $j = 1, \dots, M_k$, and the weights w_k and nodes u_k are derived using the generalized Gaussian quadrature as discussed in [16]. In general, the total number of exponential terms $S_{exp} = \sum_{k=1}^{s(\varepsilon)} M_k$ is less than the number of terms in the multipole and local expansions. The linear mapping from O_n^m to $W(k, j)$ is referred to as the \mathcal{T}_{ME} translation operator.

Notice that shifting the exponential expansion centered at the origin to the new center $X_0 = (x_0, y_0, z_0)$ of an interaction list box is “diagonal”, as described by the “exponential-to-exponential” (\mathcal{T}_{EE}) operator as follows:

Definition 5 (\mathcal{T}_{EE} Translation) For the exponential expansion in Eq. (12) of box c centered at the origin and a box in the interaction list centered at $X_0 = (x_0, y_0, z_0)$, the shifted exponential expansion for any point (x, y, z) is given by

$$\sum_{k=1}^{s(\varepsilon)} \sum_{j=1}^{M_k} V(k, j) \cdot e^{-(u_k + \lambda)(z - z_0)} \cdot e^{i\sqrt{u_k^2 + 2u_k \lambda} \cdot ((x - x_0) \cdot \cos(\alpha_{j,k}) + (y - y_0) \cdot \sin(\alpha_{j,k}))},$$

where

$$V(k, j) = W(k, j) \cdot e^{-(u_k + \lambda)z_0} \cdot e^{i\sqrt{u_k^2 + 2u_k\lambda} \cdot (x_0 \cdot \cos(\alpha_{j,k}) + y_0 \cdot \sin(\alpha_{j,k}))}$$

for $k = 1, \dots, s(\varepsilon)$, $j = 1, \dots, M_k$. The linear operator mapping the coefficients $\{W(k, j)\}$ to the coefficients $\{V(k, j)\}$ is denoted by \mathcal{T}_{EE} . As the corresponding matrix is diagonal, \mathcal{T}_{EE} is referred to as a diagonal operator.

Once a box collects all the exponential expansions from its interaction list boxes, the exponential expansion can be translated to a local expansion using the ‘‘exponential-to-local’’ operator as follows:

Definition 6 (\mathcal{T}_{EL} Translation) Suppose that the potential for $X = (x, y, z)$ (collected from the ‘‘upward’’ exponential expansions) is given by

$$\Phi(X) = \lambda \sum_{k=1}^{s(\varepsilon)} \sum_{j=1}^{M_k} W(k, j) \cdot e^{-(u_k + \lambda)z} \cdot e^{i\sqrt{u_k^2 + 2u_k\lambda} \cdot (x \cdot \cos(\alpha_{j,k}) + y \cdot \sin(\alpha_{j,k}))}$$

Then there exists an integer p , such that

$$\left| \Phi(X) - \sum_{n=0}^p \sum_{m=-n}^n L_n^m \cdot i_n(\lambda r) \cdot Y_n^m(\theta, \phi) \right| = O(\varepsilon), \quad (13)$$

where (r, θ, ϕ) are the spherical coordinates of X with respect to the box center and

$$L_n^m = (-1)^n i^{|m|} \sqrt{4\pi} \sqrt{2n+1} \sqrt{\frac{(n-|m|)!}{(n+|m|)!}} \sum_{k=1}^{s(\varepsilon)} P_n^{|m|}\left(\frac{u_k + \lambda}{\lambda}\right) \sum_{j=1}^{M_k} W(k, j) \cdot e^{im \cdot \alpha_{j,k}}$$

for $n = 0, \dots, p$, $m = -n, \dots, n$. The linear operator converting the coefficients $\{W(k, j)\}$ into the coefficients $\{L_n^m\}$ is denoted by \mathcal{T}_{EL} .

Note that a box c can have up to 189 interaction list boxes, assume there are p^2 terms in the multipole and local expansions, then to convert the multipole expansion of c to its interaction list boxes, $189p^4$ operations are required by the operator \mathcal{T}_{ML} . On the other hand, using the decomposition

$$\mathcal{T}_{ML} = \mathcal{T}_{EL} \circ \mathcal{T}_{EE} \circ \mathcal{T}_{ME}, \quad (14)$$

approximately $2p^4 + 189p^2$ operations are necessary. For 7 digits accuracy ($p = 21$), the new version using the diagonal translation can be orders of magnitude more efficient than the original version. Further notice that the

exponential expansions of c and its siblings can be merged before being sent to their common interaction list boxes, and one can use the “point-and-shoot” technique to reduce the $2p^4$ operations to approximately $6p^3$ or less [7], in our FMM-Yukawa solver, the original $189p^4$ operations are further reduced to approximately $40p^2 + 6p^3$. Preliminary numerical experiments show that the break even point of the new version FMM becomes 1500 for 6-digit accuracy and about 750 for 3-digit.

Instead of further discussions of the translation operators and their derivations, we refer interested readers to [8] for further details, and to [7] where the new version of FMM was discussed for the Laplace equation. We also want to mention the recent work of Boschitsch, Fenley, and Olson [3], who give (to the best of our knowledge) the first detailed description of an FMM for (1). Their scheme, however, does not take into account the plane wave representation and relies only on classical multipole and local expansions.

3 Software Structure and Installation Instructions

3.1 Algorithm Description

We present the pseudo-code to explain the structure of the algorithm.

FMM-Yukawa: an adaptive FMM solver for Screened Coulomb Interactions

Step 1: Initialization

Generate an adaptive oct-tree structure and the precomputed table of coefficients for different translation operators.

Comment [l_{max} denotes the maximum refinement level in the adaptive oct-tree structure determined by the prescribed number s representing the maximum allowed number of particles in a childless box.]

Step 2: Upward Pass

```

for  $l = l_{max}, \dots, 0$ 
  for all boxes  $j$  on level  $l$ 
    if  $j$  is a leaf node
      form the multipole expansion using Eq. (2).
    else
      form the multipole expansion by merging the expansions of its

```

```

        children using the  $\mathcal{T}_{MM}$  translation operator.
    endif
end
end
end

```

Comment [In upward pass, multipole expansions for all nodes in the oct-tree structure will be formed.]

Cost [The dominating cost in the upward pass is approximately Mp^4 (or Mp^3 if using the “point-and-shoot” technique [4]), where M is the number of nodes (boxes) in the oct-tree structure.]

Step 3: Downward Pass

initialize the local expansion for the coarsest level based on different boundary conditions.

```

for  $l = 1, \dots, l_{max}$ 
    for all boxes  $j$  on level  $l$ 
        shift the local expansion of  $j$ 's parent to  $j$  itself using  $\mathcal{T}_{LL}$  operator.
        collect interaction list contribution using the  $\mathcal{T}_{ME}$ ,  $\mathcal{T}_{EE}$ , and  $\mathcal{T}_{EL}$  translations.
    end
end
end

```

Comment [In downward pass, local expansion will be formed for each box to collect far field contributions.]

Cost [The downward pass requires approximately $6Mp^4 + 40Mp^2$ operations when the “merge-and-shift” technique is used, or approximately $20Mp^3 + 40Mp^2$ operations when the “point-and-shoot” technique is also used.]

Step 4: Evaluate Local Expansions

```

for each leaf node (childless box)
    evaluate the local expansion at each particle location.
end

```

Comment [At this point, for each charge point, its far field contribution has been computed.]

Cost [The amount of work is approximately p^2N for evaluating the local expansions at all charge locations.]

Step 5: Local interactions

```

for  $i = 1, \dots, N$ 
    compute the direct interaction of charge  $i$  with those particles in the
        neighboring boxes.
end

```

Cost [The cost of this step depends on how the charges are distributed in the computational domain. When they are approximately uniformly distributed, the total amount of work is approximately $27 \cdot s \cdot N$ where 27 is the total number of neighboring boxes, and s is the prescribed maximum number of particles allowed in each box.]

3.2 Installation instructions

After the package is downloaded and extracted to local computer, the user will find the following directories:

- doc: contains license and readme.txt.
- source: source files and makefile.

The package has been successfully compiled using the Intel©compiler for Linux and GNU©F95 compiler. The compiled executable is called “fmm” which can be changed by modifying “Makefile”. One function which may be machine dependent is the subroutine for getting the current CPU clock information for timing purposes. The users should check their computer platform and modify “second.f”.

The main driver for FMM-Yukawa is called “adapyukdriver.f”. The main function is *FMMYUK-A* in file “fmmadapyuk.f”, which calls the subroutine *D3MSTRCR* to generate the adaptive tree structure, and *YADAPFMM* for calculating the force field and potential. There are three important variables defined in the header file “parm-ayuk.h”. *NBOX* is the maximum allowed number of particles (=s) in a childless box, *NTERMS* is the number of terms in first summation of the multipole and local expansions, and *NLAMBDA* is the number of terms in the first sum of the exponential expansion. Currently only 3 digits and 6 digits accuracies are allowed. More options will be added in future versions. Input variables include the frequency *BETA*, the number of charges *NATOMS*, charge locations *ZAT(3,NATOMS)*, and the charge *CHARGE(NATOMS)* each particle carries. FMM-Yukawa will calculate and output the potential *POT(NATOMS)* and the field *FIELD(3,NATOMS)*.

3.3 A Sample Driver File

In the following, we provide a sample driver to explain how FMM-Yukawa can interface with other codes.

A driver file for FMM-Yukawa

```
IMPLICIT NONE
INTEGER *4 NATOMS,IER
PARAMETER (NATOMS=1000000)
REAL *8 BETA
REAL *8 ZAT(3,NATOMS),CHARGE(NATOMS)
REAL *8 POT(NATOMS),FIELD(3,NATOMS)
c
c—set up parameters.
c
  BETA=0.1D0
c
c—generate charges and their locations.
c
  CALL DUMMY(NATOMS,ZAT,CHARGE)
c
c—call fmm to calculate the potential and field.
c
  CALL FMMYUK_A(BETA,NATOMS,ZAT,CHARGE,POT,FIELD,IER)
c
  STOP
  END
```

4 Test run description

In this section, we present two numerical examples to show the efficiency and accuracy of our algorithm. In our simulation, we use a T9400 @ 2.53 GHz Lenovo T500 Laptop with 3 GB memory. The results of our experiments are summarized in Tables 1- 4 with all timings given in seconds and storage measured in megabytes.

In our first example, we consider the case when charges were uniformly distributed in the unit box $[-0.5, 0.5]^3$; results are reported in Tables 1-2. In our second example, we consider the case when charges were distributed randomly in the polar angles θ and ϕ on the surface of a sphere of radius 0.5, centered at the origin; results are reported in Tables 3-4. In both examples, the frequency of the equation was set to be 0.1.

For each geometry, the numerical tests were performed with three- and six-digit

accuracy. For each N tested, we run the tests with multiple choices of maximum number of charges allowed in a childless box and the optimal time observed was reported. The timings produced by the adaptive FMM algorithm were compared with those obtained by direct calculation. Because of obvious CPU considerations, it is not practical to apply the direct scheme to large-scale ensembles of particles. Thus, the direct algorithm was used to evaluate the potentials at 400 particle locations (for $N > 400$), and the resulting CPU time was extrapolated. Similarly, the accuracy of the algorithm was calculated at those 400 locations via the formula

$$E = \left(\frac{\sum_{i=1}^{400} |\Phi(x_i) - \tilde{\Phi}(x_i)|^2}{\sum_{i=1}^{400} |\Phi(x_i)|^2} \right)^{1/2}. \quad (15)$$

Table 1
Timing Results for the FMM for 3-digit Accuracy with Charges Uniformly Distributed in a Cube

N	Levels	Boxes	p	S_{exp}	Storage	T_{fmm}	T_{dir}	Error
750	2	41	9	67	0.12	0.07	0.08	$9.7 \cdot 10^{-5}$
1500	3	165	9	67	0.19	0.15	0.32	$1.7 \cdot 10^{-4}$
5000	4	649	9	67	4.60	0.40	3.4	$1.9 \cdot 10^{-4}$
10000	4	681	9	67	5.08	0.77	14	$2.6 \cdot 10^{-4}$
20000	5	4729	9	67	26.44	2.25	54	$3.6 \cdot 10^{-4}$
50000	5	4776	9	67	28.52	4.29	337	$3.5 \cdot 10^{-4}$
100000	5	4777	9	67	31.57	11.02	1349	$4.4 \cdot 10^{-4}$
200000	6	33506	9	67	184.97	18.82	5480	$4.6 \cdot 10^{-4}$
500000	6	37363	9	67	223.05	47.32	33900	$4.6 \cdot 10^{-4}$

Table 2
Timing Results for the FMM for 6-digit Accuracy with Charges Uniformly Distributed in a Cube

N	Levels	Boxes	p	S_{exp}	Storage	T_{fmm}	T_{dir}	Error
750	2	41	18	311	3.05	0.16	0.08	$5.9 \cdot 10^{-8}$
1500	3	81	18	311	3.92	0.32	0.32	$6.8 \cdot 10^{-8}$
5000	4	197	18	311	6.52	0.91	3.4	$1.1 \cdot 10^{-7}$
10000	4	681	18	311	16.76	1.71	14	$1.5 \cdot 10^{-7}$
20000	4	681	18	311	17.37	3.86	54	$1.8 \cdot 10^{-7}$
50000	5	4776	18	311	103.31	10.65	337	$2.3 \cdot 10^{-7}$
100000	5	4777	18	311	106.38	18.47	1349	$2.8 \cdot 10^{-7}$
200000	5	4945	18	311	115.94	46.27	5480	$2.5 \cdot 10^{-7}$
500000	6	35363	18	311	800.06	101.45	33862	$3.9 \cdot 10^{-7}$

Table 3
Timing Results for the FMM for 3-digit Accuracy with Charges Distributed on the Surface of a Sphere

N	Levels	Boxes	p	S_{exp}	Storage	T_{fmm}	T_{dir}	Error
750	3	74	9	67	1.40	0.09	0.08	$1.2 \cdot 10^{-4}$
1500	4	104	9	67	1.60	0.15	0.32	$8.8 \cdot 10^{-5}$
5000	5	391	9	67	3.29	0.36	3.4	$1.3 \cdot 10^{-4}$
10000	6	639	9	67	4.87	0.84	14	$1.2 \cdot 10^{-4}$
20000	7	1702	9	67	10.92	2.01	54	$1.9 \cdot 10^{-4}$
50000	9	5899	9	67	34.27	4.18	337	$1.9 \cdot 10^{-4}$
100000	10	10685	9	67	55.50	8.25	1349	$2.3 \cdot 10^{-4}$
200000	11	23044	9	67	131.33	16.13	5480	$2.2 \cdot 10^{-4}$
500000	12	57950	9	67	328.60	38.99	33862	$2.5 \cdot 10^{-4}$

Table 4
Timing Results for the FMM for 6-digit Accuracy with Charges Distributed on the Surface of a Sphere

N	Levels	Boxes	p	S_{exp}	Storage	T_{fmm}	T_{dir}	Error
750	2	65	18	311	3.55	0.17	0.08	$1.0 \cdot 10^{-7}$
1500	3	100	18	311	4.31	0.31	0.32	$7.1 \cdot 10^{-8}$
5000	5	323	18	311	9.11	0.93	3.4	$0.5 \cdot 10^{-8}$
10000	6	639	18	311	15.90	1.79	14	$8.7 \cdot 10^{-8}$
20000	7	1702	18	311	38.34	4.17	54	$1.0 \cdot 10^{-7}$
50000	7	2079	18	311	47.92	9.39	337	$1.6 \cdot 10^{-7}$
100000	9	5976	18	311	131.01	18.32	1349	$1.8 \cdot 10^{-7}$
200000	10	10987	18	311	240.03	37.81	5480	$1.7 \cdot 10^{-7}$
500000	12	32715	18	311	704.60	89.79	33862	$1.8 \cdot 10^{-7}$

We can see from Figure 1 that the actual CPU time required by the FMM-Yukawa solver grows approximately linearly with the number of particles N .

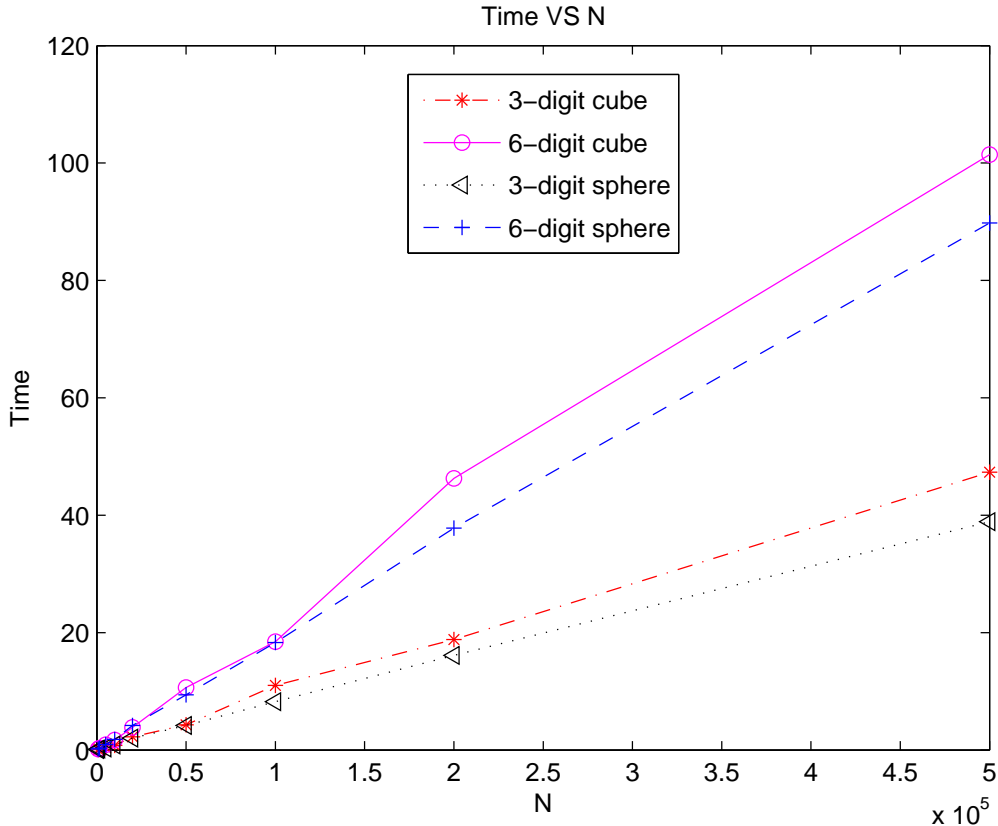


Fig. 1.

5 Conclusions and Acknowledgements

We have described a software package for evaluating the screened Coulomb interactions. It uses an adaptive new version of FMM based on a diagonal form for the multipole-to-local translation operator, which extends the “modern” FMM for the Laplace equation [4,7]. Numerical experiments show that our algorithm is asymptotically linear in operations and memory storage for nearly uniform particle distributions, with optimized prefactors. This packaged can be applied to particle method simulations and the efficient solution of integral equation methods for many physical systems.

We would like to express our gratitude to Prof. Leslie Greengard at the Courant Institute of Mathematical Sciences, Professor Vladimir Rokhlin at Yale, and members in their groups. Without their help and encouragements, this package is simply impossible. JH and BZ were supported by NSF (NSF0811130 and NSF0411920), and the NSF Center of Theoretical Biological Physics (CTBP) and Institute for Mathematics and Its Applications (IMA). JJ was supported by the Computer Science and Mathematics Division at Oak Ridge National Laboratory. Their support

is thankfully acknowledged.

References

- [1] A. W. Appel. An efficient program for many-body simulation. *SIAM Journal on Scientific and Statistical Computing*, 6(1):85–103, 1985.
- [2] J. Barnes and P. Hut. A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature*, 324(6096):446–449, December 1986.
- [3] A. H. Boschitsch, M. O. Fenley, and W. K. Olson. A fast adaptive multipole algorithm for calculating screened coulomb (yukawa) interactions. *J. Comput. Phys.*, 151(1):212–241, 1999.
- [4] H. Cheng, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm in three dimensions. *J. Comput. Phys.*, 155(2):468–498, 1999.
- [5] T. Darden, D. York, and L. Pedersen. Particle mesh ewald: An $n \log(n)$ method for ewald sums in large systems. *The Journal of Chemical Physics*, 98(12):10089–10092, 1993.
- [6] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73(2):325–348, December 1987.
- [7] L. Greengard and V. Rokhlin. A new version of the fast multipole method for the laplace equation in three dimensions. *Acta Numerica*, 6(-1):229–269, 1997.
- [8] L. F. Greengard and J. Huang. A new version of the fast multipole method for screened coulomb interactions in three dimensions. *J. Comput. Phys.*, 180(2):642–658, 2002.
- [9] A. J. Juffer, E. F. F. Botta, B. A. M. van Keulen, A. van der Ploeg, and H. J. C. Berendsen. The electric potential of a macromolecule in a solvent: A fundamental approach. *J. Comput. Phys.*, 97(1):144–171, 1991.
- [10] S. Kapur and D. E. Long. Ies3: Efficient electrostatic and electromagnetic simulation. *IEEE Comput. Sci. Eng.*, 5(4):60–67, 1998.
- [11] J. Liang and S. Subramaniam. Computation of molecular electrostatics with boundary element methods. *Biophysical Journal*, 73(4):1830–1841, October 1997.
- [12] E. T. Ong, K. H. Lee, and K. M. Lim. A fast algorithm for three-dimensional electrostatics analysis: fast fourier transform on multipoles (fftm). *International Journal for Numerical Methods in Engineering*, 61(5):633–656, 2004.
- [13] E. T. Ong, K. M. Lim, K. H. Lee, and H. P. Lee. A fast algorithm for three-dimensional potential fields calculation: fast fourier transform on multipoles. *J. Comput. Phys.*, 192(1):244–261, 2003.

- [14] J. R. Phillips, J. K. White, and A. Member. A precorrected-fft method for electrostatic analysis of complicated 3-d structures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 16:1059–1072, 1997.
- [15] D. A. S. W. B. Russell and W. R. Schowalter, editors. *Colloidal Dispersions*. Cambridge University Press, 1991.
- [16] N. Yarvin and V. Rokhlin. Generalized gaussian quadratures and singular value decompositions of integral operators. *SIAM Journal on Scientific Computing*, 20(2):699–718, 1998.