A NOVEL DIFFERENTIAL ALGEBRAIC ADAPTIVE FAST MULTIPOLE METHOD

S. Abeyratne[^], *S. Manikonda*[*], *B. Erdelyi*[*,^]

[^]Department of Physics, Northern Illinois University, DeKalb, IL 60115
[*]Physics Division, Argonne National Laboratory, Argonne, IL 60439

Abstract

The direct pairwise calculation of the potential/electric fields created by a very large number of particles is computationally impractical since it requires a long run time and a large amount of memory. The Fast Multipole Method (FMM) is a fast algorithm, which scales linearly with the number of particles and it enables highly accurate evaluation of the potentials and fields among the large number of particles using less memory compared to some other fast methods. The FMM has two main forms, non-adaptive and adaptive. The former is suitable for uniform distributions while the latter is more efficient for non-uniform distributions typically encountered in beam physics. This paper presents an implementation of a novel 3D adaptive FMM algorithm and some results obtained from simulations performed with non-uniform particle distributions.

INTRODUCTION

Pairwise calculation of Coulomb interaction is not feasible when the number of particles becomes very large since the computational complexity is of the order O(N^2). The FMM shows great promise to diminish the computational cost to O(N). The FMM can be split into two parts, data structuring and the calculation of the potential or the field. The former was discussed in previous papers [1, 2]. This paper concentrates on the implementation of the FMM to calculate the potential and fields at prescribed target points. In our analysis we use homogeneous and inhomogeneous distributions of source and target points. Therefore, instead of the regular FMM we use the adaptive FMM because it can cater to both types of distributions.

For data structuring, we have used an octree data structure, and the code is written in C++. For calculating the potential, the FMM is used and the code is written in the differential algebraic-methods-based COSY Infinity [3]. In data structuring we use certain unique capabilities available in C++ such as bit (de)interleaving and dynamic memory management, which are not available in COSY Infinity. Another advantage of C++ over COSY Infinity is that it is somewhat faster.

We measured the total run-time, which consists of the time taken for data structuring and for the calculation of the potential. The time taken for data structuring is a small fraction of the run-time, and in our previous paper we have shown that it scales linearly with the number of particles, if that number is large enough [3]. In Figure 5, we show that the total run-time for the FMM also linearly scales with the number of particles.

ALGORITHM

For details on the structure and implementation of the algorithm to create our data structures is described in [2]. When the data structure is ready, the next step is to run the FMM proper. There are two major parts in the FMM used to calculate the potential: the upward pass and the downward pass.

Upward Pass

We start the upward pass at the finest level, or the highest level, of each tree. The upward pass consists of two steps.

- Step 1: For each tree, calculate the multipole expansions of all leaf nodes at the finest level around their centers.
- Step 2: Multipole-to-multipole (M2M) shift. Advance to the next lower level or the parent level of the finest level by combining child nodes and then compute the multipole expansion of the parent box around its center by adding the multipole expansions of child nodes around their centers.

Repeat this step until level two is reached.

Downward Pass

The downward pass consists of three steps.

- Step 1: Start at level 2 and do a multipole-to-local (M2L) shift for each box. The local expansion of a particular box around its center is computed from the multipole expansions of the boxes belonging to the interaction list of that box.

In our results, we noticed that this transformation takes a considerably longer amount of time compared to other steps in the algorithm. By rotating the coordinate system of the box in such a way that the z-axis of the box aligns with the line joining the centers of the box and the interaction list boxes, the 3D transformation of the M2L becomes a 1D transformation. Hence, we can significantly reduce the M2L transformation time.

- Step 2: Move to the next finer level or to the level of child nodes and compute the local expansion of child nodes around their centers from the multipole expansions of the boxes belong to the interaction list of the child nodes.
- Step 3: Re-expand the result in step 1 and compute the contribution from the local expansion of the parent box, which is known as the local-to-local (L2L) shift, and add that result to the result

ISBN 978-3-95450-122-9

obtained in step 2.

The above three steps should be repeated until the coarsest level, level two, is reached.

With the completion of the above described upward pass and downward pass, all target boxes at the finest level have their local expansions due to all the sources outside the neighborhood of each target box.

Final Summation

- Step 1: Point-to-point (P2P) calculation. In order to get the potential at targets, evaluate the local expansion around the center of each target box at each target point in the box.
- Step 2: To get the total potential, calculate the potential at each target due to the sources in the neighborhood and add it to the potential evaluated in step 1.

Also, calculate the electric fields at each target by taking the derivative of the potential at the point.

We note that all steps, except the point-to-point piece of step 1 in the final summation, are performed automatically by differential algebraic methods as detailed in [3].

EXAMPLES

As examples, we present some cases of source points generated by three different distributions: uniform, single 3D Gaussian, and nine 3D Gaussians. We placed the target points in a uniform two dimensional grid spanning [-3, 3] in x and y directions, with z=0 (a cross-section centered on the source distributions). We assumed, for simplicity, that each target and source particle carries a unit charge.

The number of sources and targets are 45000 and 2401, respectively. The order of FMM in all three examples is 9.

Example 1

The source points are uniformly distributed in a cube in the range [-1, 1] in all three directions x, y, z. The potentials were calculated in both FMM and point-to-point, or direct methods. Figure 1 shows that the relative errors of the FMM method is very small and therefore, can replace the direct method for most practical applications. The potential determined from the FMM is plotted in Figure 2.

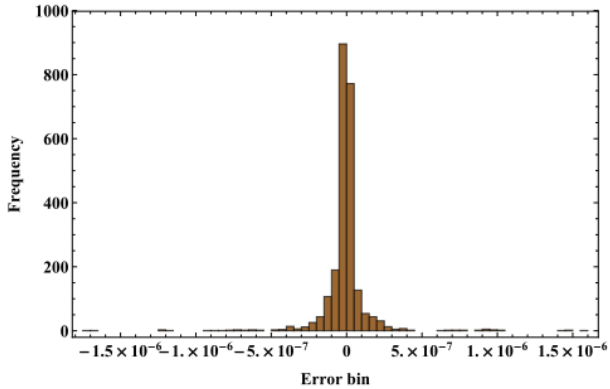

Figure 1: Histogram of the relative errors of the FMM calculated for the potential due to uniformly distributed sources in a cube.

Example 2

Source points are distributed as a single round 3D Gaussian. Targets were in the same uniform grid as the previous example. The potential is plotted in Figure 3.

Example 3

Source points are from a distribution of nine 3D Gaussians. Again, targets were on the same uniform grid. The 3D potential is plotted in Figure 4. The error histograms for examples 2 and 3 are very similar to Figure 1.

By varying the particle number N, the run time is measured, under optimized conditions specific to the data structures, distributions and order used, for Gaussian and uniform distributions using the FMM at order 2. Both types of distributions show that the run time is linear with the particle number N.

The run time for Gaussian distribution is slightly higher than that of the uniform distribution.

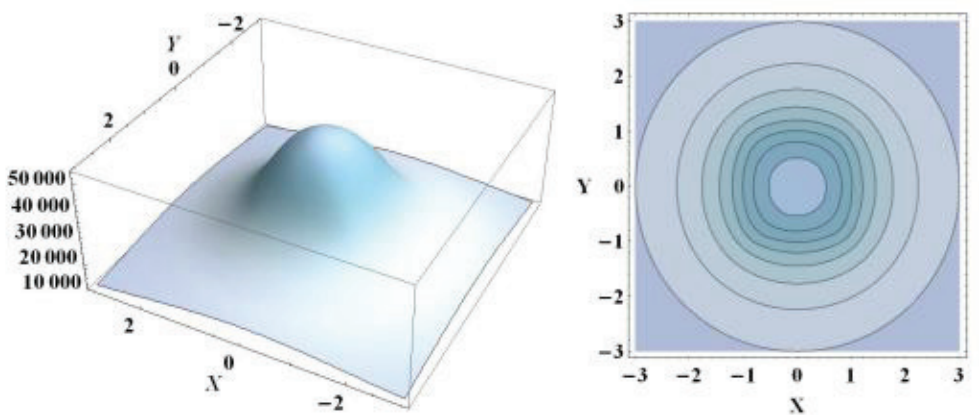

Figure 2: 3D plot (on the left) and contour plot (on the right) of potential (a.u.) due to uniformly distributed sources in a cube.

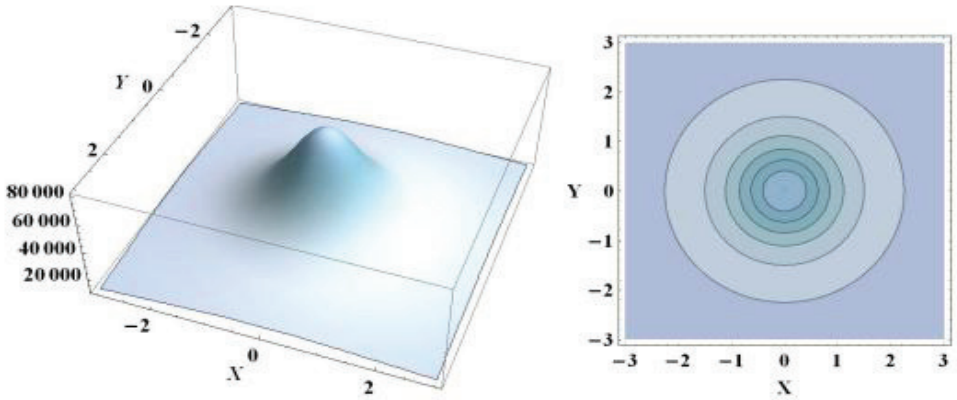

Figure 3: 3D plot (on the left) and contour plot (on the right) of potential (a.u) due to sources distributed as a single round Gaussian.

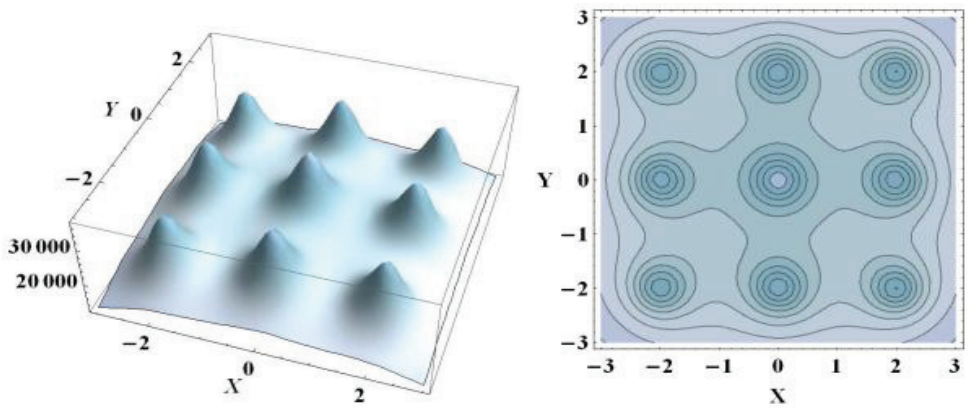

Figure 4: 3D plot (on the left) and contour plot (on the right) of potential (a.u.) due to sources distributed as a grid of nine 3D Gaussians.

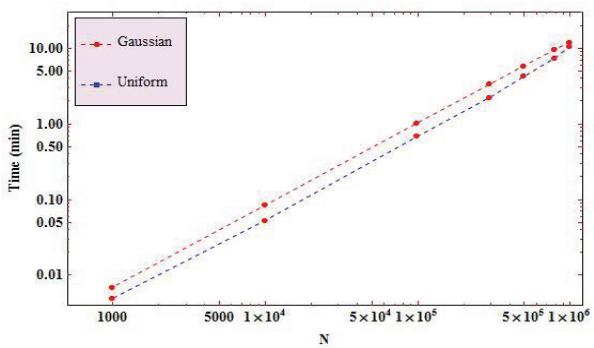

Figure 5: Run-time measured for different particle numbers N at the FMM order 2.

SUMMARY

We have implemented the FMM to calculate the potential generated by three types of source particle distributions at uniformly distributed targets on a 2D grid.

The error histogram shows that the potential calculated using the FMM agrees very well with the potential calculated using the direct pairwise method. A higher FMM order produces more accurate calculations. In addition, as the name implies, the FMM calculation is fast since the FMM runtime is significantly less than that of the direct method. In Example 1, at the optimum q value for N=8000, the measured runtime for the FMM is 0.73 minutes while that for the direct method is 2.68 minutes. Therefore, we have shown that the FMM can be utilized for accurate and fast potential/field calculation irrespective of the type of particle distributions.

ACKNOWLEDGEMENT

This work was supported by the U.S. Department of Energy, Office of Nuclear Physics, under Contract No. DE-SC0008588.

REFERENCES

[1] Nail A. Gumerov and Ramani Duraiswami, "*Fast Multipole Methods for the Helmholtz Equation in Three Dimensions*", Elsevier, 2004.

[2] S. Abeyratne, S.Manikonda and B. Erdelyi, "*Optimal Fast Multipole Method Data Structures*", Conf. Proc. C1205201 (2012) 352-354.

[3] H. Zhang and M. Berz, "*The Fast Multipole Method in Differential Algebra Framework*", Nuclear Instruments and Methods, A645 (2011) 338-344.

ISBN 978-3-95450-122-9