

THE MULTILEVEL SUMMATION METHOD: AN INTRODUCTORY LOOK AND PYTHON EXPERIMENTS (DRAFT 2)

JOSH BEVAN *

1. Problem Statement. There are many physical phenomena that fit into the category of n -body problems, systems where there are long-range pairwise interactions between each body. These include areas like molecular dynamics, electro-magnetics, gravitation, and fluid flow. The simulation of these phenomena can be computationally intensive if all pairwise interactions are calculated directly. Direct calculation requires calculations that scale as $\mathcal{O}(n^2)$. For many such simulations of practical consequence n can be on the order of millions, rendering direct approaches useless.

Fast methods seek to reduce the computational cost to $\mathcal{O}(n \log n)$ or even $\mathcal{O}(n)$ by replacing direct evaluation of long-range interactions with approximations. The two sets of method choices can be roughly lumped into two categories: periodic and non-periodic. Periodic approaches include techniques like Particle Mesh Ewald (PME) which can employ periodic basis functions, but place periodicity requirements on the solution domain. The other set of methods include tree codes and most famously the Fast Multipole Method (FMM). In contrast to PME etc. the FMM is applicable for non-periodic domains.

In contrast to these examples the Multilevel Summation Method (MSM) has several advantages. Multipole methods don't have explicit continuity of the potential in the transition from the near-field to the regions well separated from the target. In comparison to PME, MSM benefits from exact calculation of short-range interactions as well as naturally separating the potential into different length scales which more readily permits multiple time stepping.[2]

The following section presents the MSM in greater detail, though it is by no means comprehensive. The intent is to highlight the salient features of the MSM and then present some experiments in Python that implement/illustrate the concepts.

2. Approach. In many cases the potential to be calculated can be expressed as:

$$(1) \quad E(x) = C \sum_{i=1}^N \sum_{j=1}^N q_i q_j k(r_i, r_j)$$

where q represents the strength of some property (e.g. charge, mass) and r is the position vector for a given particle.

The function $k(r_i, r_j)$ is the interaction kernel, in many cases the Green's function that describes the fundamental behavior of solutions to the governing equation. For electrostatics and gravitation it is simply $1/r$. Issues arise if the kernel is simply used without accounting for the difficulty in approximating the near singular parts. Approximations for long-range interactions depend upon the availability of good approximations. Therefore a splitting is necessary to be able to separately manipulate only the smooth components.

We can rewrite the kernel as a sum of kernels of increasing reach and reducing

*jjbevan2@illinois.edu

variation. Take for example a particular splitting case[3] with four terms:

$$(2) \quad \frac{1}{z} = f_0(z) + f_1(z) + f_2(z) + f_3(z)$$

where we have made the substitution $z = |r - r'|$ and we assume the kernel functions under consideration is radial, that is $k(r, r') \rightarrow f(|r - r'|) = f(z)$.

The shortest range part is f_0 on the individual particle level, and is directly calculated. The other three parts are approximated by an interpolation on a hierarchy of increasingly coarser grid. This means pairwise interactions are not dramatically reduced to only short-range interactions. A splitting parameter a is used to control the spatial reach of each of the coarser kernels. In this example then the other terms would have ranges a , $2a$, and $4a$.

This leaves the issue of actually constructing each of these kernels. Consider an unknown smoothing function $g_a(r, r')$, with the property:

$$(3) \quad g_a(r, r') = \frac{1}{z} \text{ for } z > a$$

while for $z \leq a$ there is less variation compared to the original potential.

If this smoothing function is used one receives a telescoping sum that satisfies the original equation:

$$(4) \quad f_0(z) = 1/z - g_a(r, r')$$

$$(5) \quad f_1(z) = g_a(r, r') - g_{2a}(r, r')$$

$$(6) \quad f_2(z) = g_{2a}(r, r') - g_{4a}(r, r')$$

$$(7) \quad f_3(z) = g_{4a}(r, r')$$

Unsurprisingly the smoothness of the kernel is dependent on the smoothing function. Ideally a given g_a should be selected to be within the range of interpolation on the grid. It is worth noting that so far no approximations have been made, merely decomposed the original kernel into a more suggestive form. The actual approximations are made during the interpolation of the smooth kernels to the hierarchical grids; typically with a nodal basis defined on the grid. As an example the first interpolation of the first smooth kernel would have the form:

$$(8) \quad g_a(r, r') \approx \sum_n \sum_m \phi_n(r) g_a(r_n, r_m) \phi_m(r')$$

where r_n/r_m are grid points and ϕ_n/ϕ_m are the nodal basis functions.

We can now substitute this approximation expression into our pairwise interaction problem to yield:

$$(9) \quad C \sum_{i=1}^N \sum_{j=1}^N q_i q_j g_a(r_i, r_j) \approx C \sum_{i=1}^N \sum_{j=1}^N q_i q_j \sum_n \sum_m \phi_n(r) g_a(r_n, r_m) \phi_m(r')$$

however our nodal basis functions are compactly supported on just the grid, so the two sets of sums reduce to:

$$(10) \quad = C \sum_n \sum_m q_n^h q_m^h g_a(r_n, r_m)$$

where we now have approximate grid charge values that are the interpolation of the underlying particles:

$$(11) \quad q_n^h = \sum_{i=1}^N q_i \phi_n(r_i)$$

The result is the reduction from pairwise particle interactions to pairwise grid interactions. One interesting feature of this is that even if the particles may move, the interpolated grid locations used in the decomposition don't.

For coarser levels, g_{2a} for example, interpolation is again performed but with larger mesh spacing (double in the case of g_{2a})

$$(12) \quad C \sum_{i=1}^N \sum_{j=1}^N q_i q_j g_{2a}(r_i, r_j) = C \sum_{i=1}^N \sum_{j=1}^N q_n^{2h} q_m^{2h} g_{2a}(r_{2n}, r_{2m})$$

We now have all the parts necessary to compute the potential field E . For the finest level, compute:

$$(13) \quad e_{short} = K(x)q$$

where K is matrix constructed from the shortest range kernel f_0 and q are the particles that directly participate in the short-range interactions. Moving up the hierarchy:

$$(14) \quad q^h = (I_h)^T q \quad e_{short}^h = K_h q^h$$

$$(15) \quad q^{2h} = (I_{2h}^h)^T q \quad e_{short}^{2h} = K_{2h} q^{2h}$$

$$(16) \quad q^{4h} = (I_{4h}^{2h})^T q \quad e_{short}^{4h} = K_{4h} q^{4h}$$

and then back down

$$(17) \quad e^{4h} = e_{short}^{4h}$$

$$(18) \quad e^{2h} = e_{short}^{2h} + I_{4h}^{2h} e^{4h}$$

$$(19) \quad e^h = e_{short}^h + I_{2h}^h e^{2h}$$

We now have all components needed to calculate E :

$$(20) \quad \frac{1}{2} q e_{short} + q I_h e^h$$

Figure 1 diagrammatically shows the cycle following these steps. For the particle and grid levels and the calculation of each level's component contribution.

3. Numerical Results. The smoothing operation has been implemented for the solver to be used in the splitting. An example comparison of the pre/post-smoothing is shown in Figure 2. The routines to generate interpolation operators are mostly done as well. Time permitting additional smoothing routines will be added so that several can be compared to determine their comparative impacts.

The kernel matrix generation routines still need to be completed and the main driver script that cycles up and back down the MSM needs to have either a recursive setup, or for simplicity a fixed number of levels hard coded. Any numerical investigations will have a gentle number of points, so the hierarchy won't need to be particularly deep.

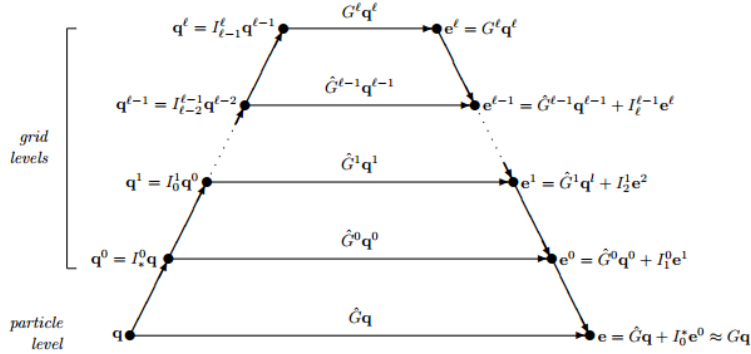


FIG. 1. Hierarchical decomposition of potential into two parts: close range interaction and smooth long-range interactions which can be approximated by an interpolation.[3]

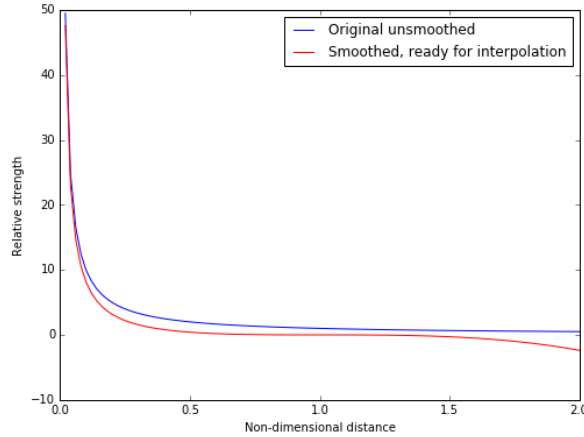


FIG. 2. Comparison of original and smoothed interaction for a single particle.

4. Conclusions. The multilevel summation method provides an alternate means of efficiently evaluating n-body problems. Also, in comparison to a FMM there are fewer components necessary for an efficient implementation. The underlying algorithm and most of the implementation details are now understood for this project and the remaining parts of the solver will be finished. A suitable set of tests will be performed to evaluate performance, contingent upon the currently unknown efficiency/robustness of this implementation.

REFERENCES

- [1] David J. Hardy, Zhe Wu, James C. Phillips, John E. Stone, Robert D. Skeel, and Klaus Schulten. Multilevel Summation Method for Electrostatic Force Evaluation. *Journal of Chemical Theory and Computation* 2015 11 (2), 766-779 DOI: 10.1021/ct5009075
- [2] Hardy, D. J. Multilevel summation for the fast evaluation of forces for the simulation of biomolecules. Ph.D. thesis, University of Illinois at UrbanaChampaign, Champaign, IL, 2006; Also Department of Computer Science Report No. UIUCDCS-R-2006-2546, May 2006.
- [3] Bond, Stephen D. Multilevel summation methods for efficient evaluation of long-range pairwise interactions in atomistic and coarse-grained molecular simulation. No. SAND2014-0355. San-

- 133 dia National Laboratories (SNL-NM), Albuquerque, NM (United States), 2014.
 134 [4] Skeel, Robert D., Ismail Tezcan, and David J. Hardy. "Multiple grid methods for classical molec-
 135 ular dynamics." *Journal of Computational Chemistry* 23, no. 6 (2002): 673-684.