The Journal of Chemical Physics

Atomic level simulations on a million particles: The cell multipole method for Coulomb and London nonbond interactions

Cite as: J. Chem. Phys. **97**, 4309 (1992); https://doi.org/10.1063/1.463935
Submitted: 06 March 1992 . Accepted: 02 June 1992 . Published Online: 31 August 1998

Hong-Qiang Ding, Naoki Karasawa, and William A. Goddard

View Online

Export Citation

Atomic level simulations on a million particles: The cell multipole method for Coulomb and London nonbond interactions

Hong-Qiang Ding, Naoki Karasawa, and William A. Goddard III
Materials Simulation Center, Beckman Institute, California Institute of Technology, Pasadena, California 91125

(Received 6 March 1992; accepted 2 June 1992)

The N^2 computations implicit in the Coulomb and other long range interactions remain the critical bottleneck in atomic-level simulations of the structure and dynamics of large systems. We report here the cell multipole method which scales linearly with N and requires only modest memory. To demonstrate the feasibility of this approach, we report systematic calculations on realistic polymer systems with up to 1.2 million atoms on a laboratory workstation. The method becomes faster than the exact method for systems of 300 atoms, and for a 1.2 million-atom polymer, it is 2377 times faster. The method treats a class of interactions of the form q_iq_j/r_{ij}^p, which includes Coulomb ($p=1$), London dispersion ($p=6$), or shielded Coulomb ($p=2$) interactions. This method is well suited for highly parallel and vector computers.

I. INTRODUCTION

In recent years, there has been a lot of progress in simulating the atomic level structure and dynamics of large molecules, with calculations on systems with thousands of atoms.[1–4] However, current methods remain inadequate to simulate the millions of particles required for interesting properties of many important systems. For example, amorphous polymers may have segments each with 100 000 atoms which associate to form partially crystalline lamellae, random coil regions, and interfaces between these regions, each of which may contribute special mechanical and chemical properties to the system. In order to carry out simulations for such systems, it is essential (i) to eliminate computational steps or storage that depend quadratically (N^2) or worse on size; (ii) to construct algorithms that scale linearly in N and have minimal overhead. In this paper we focus on the biggest bottleneck obstructing atomic-level simulations on superlarge systems, the long ranged nonbond interactions: the Coulomb interaction q_iq_j/r_{ij}, the shielded Coulomb interaction q_iq_j/r_{ij}^2 (which often replaces the Coulomb interactions to account approximately for solvent effects in biological molecules), and the London dispersion $\lambda_i\lambda_j/r_{ij}^6$. The cell multipole method (CMM) treats all these as special cases of

$$V=\sum_{i>j} q_iq_j/|\mathbf{r}_i-\mathbf{r}_j|^p, \tag{1}$$

so that one program handles any interaction in this class. The method divides space into cubic cells and use charge, dipole and quadrupoles (multipoles) to represent the cells. The computational procedures proceed quite naturally with physical intuitions transparent in each step, as clearly shown in Sec. II.

An important refinement in calculating the multipole interactions due to far-away cells is to group cells into progressively larger cells as distance increases (Sec. III). This is based on the general approach[5–7] of constructing a hierarchy of rectangular cells (many levels of cells) which facilitates the use of cells with varying sizes. The hierarchical approach was first implemented[5] as an important application of tree structures to the gravitation problem. It reduces the computations to order $N\log(N)$. The algorithm is simplified[6] by using cubic boxes and becomes more efficient. To further reduce the computation to order N, a local Taylor expansion is introduced[7] for a two-dimensional problem (using complex variables so that the expansion series is particularly simple). These adoptive tree-structured methods have been applied mostly in gravitational problems, which typically lack intrinsic scales and often exhibit diverse structures and large density fluctuations.

In contrast, molecular systems and crystals typically have clear length scales, and their structures are more stable and the density fairly uniform. These characteristics suggest that uniform cells should be adequate. We use directly charges, dipoles, and quadrupoles, etc., which have clear physical content and are easy to manipulate. The resulting CMM algorithm therefore has a much simpler structure and is capable of treating all interactions with a single program. Independent developments for the three-dimensional problem have been reported,[8] but the use of spherical harmonics or partial derivatives makes them rather complex.

The CMM method is very efficient in execution. For a typical system with 1800 atoms, it is 6 times faster than the exact algorithm which computes $N(N-1)/2$ pairs. The overhead of the method is very small: the threshold when the CMM becomes equally fast as the exact algorithm is 300 atoms. This small overhead, together with the linear dependence of N, makes this method efficient enough to rapidly calculate the nonbond interactions for very large systems. We illustrate this point with calculations on realistic polymer systems with up to a million atoms [the β-alanine starburst dendrimer[3] and poly(vinylidene fluoride)]. For the 1.2 million atom case, CMM is a factor of 2377 times faster than the exact method (see Fig. 1). Furthermore, the method gives very good accuracy: the total energy error is about 0.02% and the root-mean-square

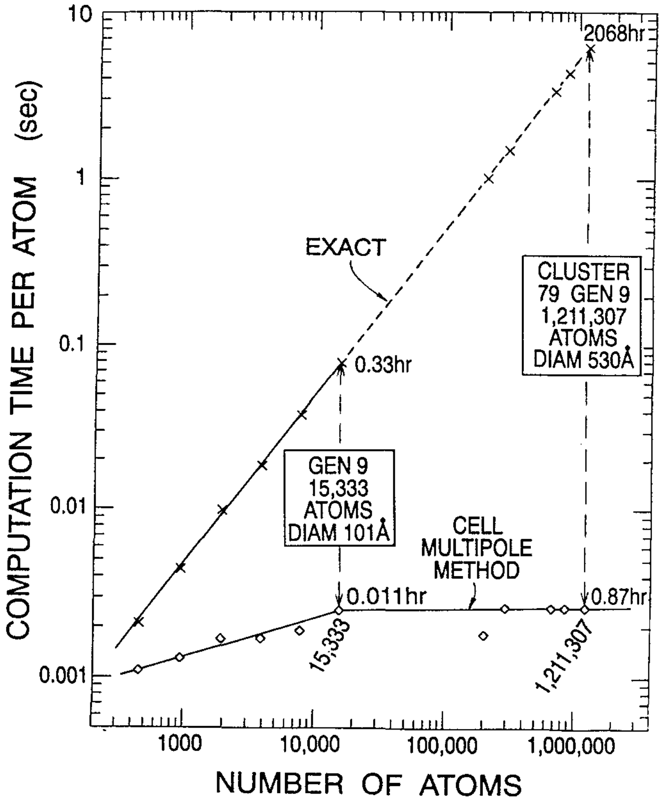

FIG. 1. CPU time per atom for calculating the Coulomb nonbond interactions for a realistic β-alanine starburst dendrimers. A dendrimer has an ammonia core with mononer units attached to the nitrogens. Additional layers of monomer units can attach to the nitrogens of monomers such that the dendrimer grow much like a tree (in contrast to the single chain polymer). Dendrimer has a sphere shape with a natural progression in size. Thus generations 4, 5, 6, 7, 8, 9 lead to 453, 933, 1893, 3813, 7653, and 15 333 atoms, respectively. For larger systems, we combine 9th generation dendrimers. Thus a closest packing sphere of 13 leads to 199 329 atoms (diameter of 244 Å) and a packing of 79 leads to 1 211 307 atoms (diameter of 530 Å). The times for the exact method for systems of 199 329 atoms or more are extrapolations.

(rms) force error is about 0.4% (at the quadrupole level), and the accuracy can be further improved systematically, as shown in Sec. IV.

II. THE CELL MULTIPOLE METHOD

We describe here a simple version of the method, in which the physical motivations are best explored and many of the essential features apparent. Consider the million atom system. The minimum cubic spatial box which holds all the atoms is used as the computational box. (The system could have any shape; using cubic box is for convenience.) Dividing the box into small cells and using charge, dipole, quadrupole to represent them, the implementation is rather natural and straightforward. The key steps are as follows:

(a) *Divide space into uniform cells.* The box containing all the atoms is divided into M equal-sized cubic cells [Fig. 2(b)]. A single pass through all atoms generates a doubly linked list which stores the grouping information (e.g., which atoms belong to a particular cell).

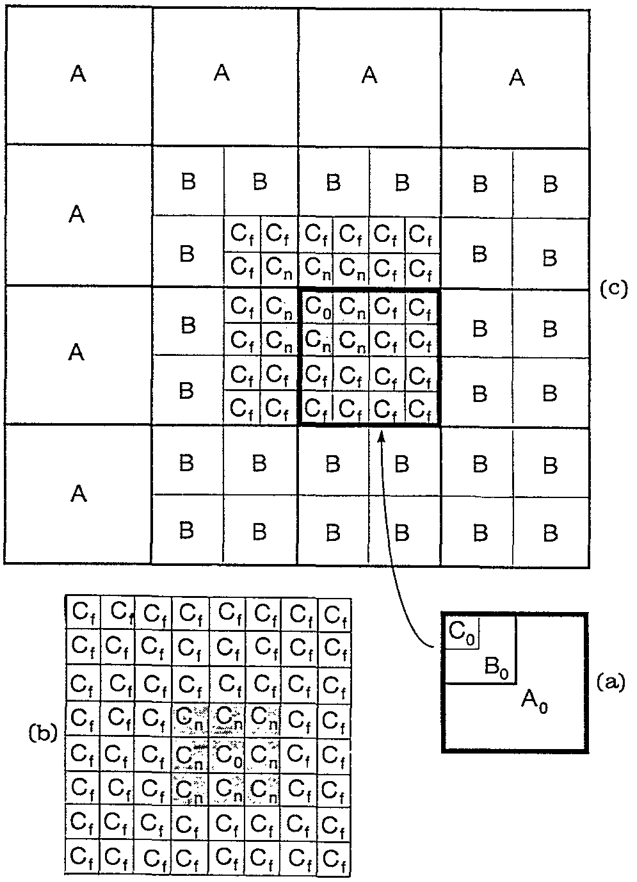

FIG. 2. Cell hierarchy. All cells are 3D cubes. (a) Indicates the hierarchy of cells. (b) Shows the deepest level cells. It also indicates the near cells and far cells. (c) The overall picture for the cell C_0. All A, B, C_f cells are far cells for C_0. Note that an A cell groups 64 C_f cells, so the cell hierarchy reduces for C_0 the number of far cells from 4069 to 415. See text for more details.

(b) *Compute multipole moments for each cell.* The interactions of each cell with any other atom outside the cell is represented by the multipole series expansion:[9]

$$V_A^{\text{pole}}(\mathbf{r})=\frac{Z}{R^p}+\frac{\mu_\alpha R_\alpha}{R^{p+2}}+\frac{Q_{\alpha\beta}R_\alpha R_\beta}{R^{p+4}}+\frac{O_{\alpha\beta\gamma}R_\alpha R_\beta R_\gamma}{R^{p+6}}+\cdots, \qquad (2)$$

where $\mathbf{R}=\mathbf{r}-\mathbf{r}_A$, $\mathbf{r}_A$ is the center of the cell A, $\mathbf{r}$ is any observer outside the cell, $\alpha=x,y,z$, and the summation over repeated index $\Sigma_{\alpha\alpha}$, etc., is implicit. The lower order moments are the charge $Z=\Sigma_i q_i$, dipoles $\mu_\alpha=p\Sigma_i q_i r_{i\alpha}$, quadrupoles

$$Q_{\alpha\beta}=\sum_i q_i \frac{p}{2}\,[(p+2)r_{i\alpha}r_{i\beta}-\delta_{\alpha\beta}r_i^2],$$

and octopoles,

$$O_{\alpha\beta\gamma}=\sum_i q_i \frac{p(p+2)}{6}\,[(p+4)r_{i\alpha}r_{i\beta}r_{i\gamma}-(r_{i\alpha}\delta_{\beta\gamma}+r_{i\beta}\delta_{\gamma\alpha}+r_{i\gamma}\delta_{\alpha\beta})r_i^2],$$

etc., where $r_{i\alpha}$ is the α component of the position vector for atom i measured with respect to the center of the cell A.

(c) *Partition the interactions into near fields and far fields.* The multipole series is essentially an expansion in terms of d/R, where d is the cell size. Thus, the cell–cell and cell–atom interactions can be computed using the multipole expansion, as long as the distances are at least one unit away, i.e., $d/R < 1/2$. Thus for each cell [C_0 in Fig. 2(b), e.g.], we classify all cells into two classes, the 27 *nearby* cells which includes the 26 neighbor [C_n in Fig. 2(b)] and the cell itself, and the far-away cells which include all other cells.

Consider an atom i in cell C_0. The interaction with any atoms in those far-away cells are computed via the multipole expansion, which will be referred to as far fields. The interactions with atoms in the 27 nearby cells cannot be computed through the multipole expansion because they are too close by; thus we compute them directly, pair by pair. Fortunately, the number of atoms in these 27 nearby cells is small ($27N/M$). These interactions will be referred to as near fields. Hence potential is decomposed as

$$V(\mathbf{r}_i)=V_{\rm far}(\mathbf{r}_i)+V_{\rm near}(\mathbf{r}_i),$$

or more explicitly,

$$V(\mathbf{r}_i)=\sum_{A\in{\rm far}} V_A^{\rm pole}(\mathbf{r}_i-\mathbf{r}_A)+\sum_{j\in{\rm near}}\frac{q_j}{|\mathbf{r}_i-\mathbf{r}_j|^p}, \qquad (3)$$

where the sum over A is through all far cells [C_f in Fig. 2(b)].

(d) *Convert multipole fields to Taylor coefficients.* On average, there are N/M atoms in cell C_0. Instead of repeating the sum over all far cells C_f for each atom i to compute $V_{\rm far}(\mathbf{r}_i)$, we notice that these atoms are rather close-by relative to those far-away cells and thus the interactions generated by those far-away cells vary little from one atom position to another for these atoms inside C_0. Therefore, we express $V_{\rm far}(\mathbf{r})$ as a local Taylor series expansion about the center of C_0,

$$\sum_A V_A^{\rm pole}(\mathbf{r}-\mathbf{r}_A)=V^{(0)}+V_\alpha^{(1)}r_\alpha+V_{\alpha\beta}^{(2)}r_\alpha r_\beta+\cdots$$

$$\equiv V^T(\mathbf{r}), \qquad (4)$$

where both the atom position $\mathbf{r}$ and the cell position $\mathbf{r}_A$ are with respect to the center of cell C_0. In this way, we need only compute the Taylor series coefficients once and simply evaluate them for each of the N/M atoms in the cell C_0. This significantly speeds up the computation.

The Taylor coefficients are computed by expanding each multipole term in $\mathbf{r}$. For example, the charge term is expanded as

$$\frac{Z}{|\mathbf{r}-\mathbf{r}_A|^p}=\frac{Z}{r_A^p}+\frac{pZ\mathbf{r}_A}{r_A^{p+2}}\cdot\mathbf{r}+\cdots,$$

the dipole term as

$$\frac{\mu\cdot(\mathbf{r}-\mathbf{r}_A)}{|\mathbf{r}-\mathbf{r}_A|^{p+2}}=\frac{-\mu\cdot\mathbf{r}_A}{r_A^{p+2}}$$

$$+\left[\frac{\mu}{r_A^{p+2}}-\frac{(p+2)(\mu\cdot\mathbf{r}_A)\mathbf{r}_A}{r_A^{p+4}}\right]\cdot\mathbf{r}+\cdots\cdots,$$

etc. $V^{(0)}$ is then the sum of all constant terms. $V^{(1)}$ is the sum of the linear coefficients of $\mathbf{r}$, etc. This is repeated for all far cells. Thus $V_{C_0}^T(\mathbf{r})$ contains interactions due to *all* atoms except those in the 27 near cells which are evaluated as in Eq. (3). The derivatives of the Taylor series give the forces, the Hessian, etc. This completes the simple version of the multipole method.

The memory usage of the method is modest. Storing multipoles, Taylor coefficients, and cell indices, we need $104M+8N$ bytes (M=total number of cells). Plus $28N$ memory for coordinates, charges and forces, and assuming $M\simeq N/4$ (optimum choice), the total memory is $\sim 60N$ bytes.

We note that many molecular systems in simulations are not cubic like, rather they have shapes more similar to spheres. This implies that many cells in the cubic space will be empty. Plus some density fluctuations, the number of empty cells is typically about 25% in the systems we have calculated. This causes no waste of computation, however, because each cell has a number count and the computer program checks this count each time it starts with a cell and skips the cell if it is empty. The amount of computer memory for these empty cell is about 10% of the total memory requirement. An alternative approach uses a flexible memory allocation scheme, which will disrupt the streamlined structures here, losing some of the efficiency of this algorithm.

A particular advantage of this simple algorithm is that the far fields represented by the Taylor series change very little during dynamics or minimization. Thus one can use the same Taylor coefficients (without updating) for a number of dynamical time steps (say 100). We have tested this approach for calculations on polymers and find that, even for updating every 100 time steps, the force errors remain essentially the same as those incurring in recomputing the Taylor coefficients anew each time step (for example, the rms force errors increase from 0.38% to 0.40%). This reduces the computation to essentially those of nearby atoms. Thus the total computation is essentially linearly proportional to the total number of atoms. For not very large N, this is a rather efficient method. (The timings reported in this paper, however, do not incorporate this advantage.)

III. THE HIERARCHICAL REFINEMENT

The procedure (d) in the above method, the multipole field to Taylor series conversion, can be calculated much more efficiently by grouping cells into larger and larger cells as distance increases. We first explain the necessity of the grouping and introduce a hierarchy of cells as a device for the grouping. The calculation then proceeds on the second level, level by level, down to the deepest level.

A drawback of the above described simple version of the method (Sec. II) is the calculation of the $V_{\rm far}^T$ which involves the sum over $M-27$ far cells [step (d)]. Since M is proportional to N (the optimal average number of atoms per cell is $N/M\simeq 4$), the computation of $V_{\rm far}^T$ for all cells is about $M(M-27)\simeq N^2/16$, which is about 16 times faster than the direct sum over each individual atoms in all the far cells. This far-fields calculation is typically repeated every 100 or 200 time steps. For a few thousands atoms or less, the $N^2/16$ calculation every 100 steps is a reasonable overhead of the simple method. But, as the number of atoms further increase, the quadratic increase of $N^2/16$ quickly goes out of scale, even if it is evaluated every 100 time steps. For the million atoms we have in mind, this calculation is impossible. Therefore, the simple version of the cell multipole method is not suited for large number of atoms, although it sets up the basis for further significant refinement.

Careful examination of summation process in calculating V_{far}^T reveals that the straightforward summation over $M-27$ far cells is very inefficient, because the interactions from distant cells (with small d/R, the cell size to distance ratio) are computed much more accurately than the interactions from not-so-distant cells (with larger d/R). Since the errors are determined by the largest d/R cases, the summation over the distant cells are thus computed with an accuracy unnecessarily high. This waste of accuracy is further deteriorated because the number of distant cells grows as R^2, i.e., we compute increasingly more and more cell interactions with accuracies increasing unnecessarily higher and higher. There are several ways one can correct this waste of computation. The simplest one is to calculate the distant cells less and less accurate, by neglecting the octopole terms, the quadrupole terms, the dipole terms as the distance increases. But the charge terms must be retained at least. Hence this scheme reduces the calculation only by a constant factor. A different way, first proposed by Appel[5] as an application of the tree structure (computer science) to the astrophysics problem and later refined by Barnes and Hut[6] and by Greengard and Rokhlin,[7] is to group small cells into larger cells. As the distance increases, so does the cell size, such that the cell size to distance ratios remain approximately constant. Thus the accuracy of the multipole calculation remains approximately the same for all cells, both the closer-smaller cells and the distant-larger cells. To implement this scheme, a hierarchy[5-7] of cells is introduced as a bookkeeping method and the detailed procedures involve modification of procedures (a), (b), and (d), as described in the following.

(a1) *Dividing space into a hierarchy of uniform cells.* We introduce a family of cells, as shown in Fig. 2. The entire (cubic) space is considered a cell at level 0. This cell is divided into 8 children cells (level 1) each of which is further divided into 8 grandchildren (level 2), etc. Finally, at the deepest level L we have $M=8^L$ cells, each with only a few (say 4) atoms. Thus, a million atoms requires $L \simeq \log_8(10^6/4) \simeq 6$ levels. Figure 2(c) shows a four-level system. The level 2,3,4 cells are denoted as A, B, C, respectively. Comparing with the simple cell approach, shown as Fig. 2(b), here we have built in some internal structure among the cells. This reduces the number of far cells dramatically. For the four-level system shown in Fig. 2, for a cell C_0, the original $16^3-3^3=4069$ far cells are reduced to 415 cells ($37A$, $189B$, $189C_f$). [For the million particle system, a level 6 cell interacts with 189 cells at each level, from level 3 through 6, and 37 at level 2, i.e., it interacts with a total of 1171 cells instead of the 262 117 original cells.]

(b1) *Compute multipoles for each cell at each level.* Multipole moments for the deepest level (C level in Fig. 2) are first calculated as in the simple method case. Each higher level cells [B_0 cells in Fig. 2(a), e.g.] contains 8 lower level children cells [C_0 cells in Fig. 2(a)]; thus its multipole moments are obtained by translating and adding those of the 8 children cells. For examples, the charge of the parent cell is a direct sum of the charges of the 8 children cells.

$$Z^{(l-1)}=\sum_{k=1}^{8}\sum_i q_{ik}=\sum_{k=1}^{8} Z_k^{(l)},$$

where Σ_k is over the eight children cells (C_0 and its 7 brothers), Σ_i is over all atoms in this kth child cell, q_{ik} is the charge of the ith atom in kth cell, and $Z_k^{(l)}=\Sigma_i q_{ik}$ is the charge of the kth cell at level l. Further, denoting $\mathbf{r}_{ik}$ as the position of charge q_{ik} with respect to the center of kth cell and $\mathbf{R}_k$ as the vector from the center of B_0 to the center of the kth cell, we can easily obtain the combination relation for the dipole of B_0 as

$$\mu^{(l-1)}=\sum_{k=1}^{8}\sum_i q_{ik}(\mathbf{r}_{ik}+\mathbf{R}_k)=\sum_{k=1}^{8}(\mu_k^{(l)}+Z_k^{(l)}\mathbf{R}_k).$$

The combination relations for quadrupoles and octopoles can be obtained similarly.

(d1) *Compute the Taylor coefficients level by level.* The conversion of multipole fields to Taylor series, as described in procedure (d), is now carried out in different sequence. Consider a particular level 4 cell, C_0, and denote its parent and grandparent as B_0 and A_0, respectively (see Fig. 2). The contributions to the Taylor series $V_{C_0}^T(\mathbf{r})$ come from the 415 far cells as explained in procedure (a1). One can sum up the contributions directly, starting with the 37 A cells, then the 189 B cells and finally the 189 C_f cells, an approach essentially similar to that of Barnes and Hut.[6] This is slower, but gives better accuracy. Here we take an indirect approach, essentially similar to that of Greengard and Rokhlin,[7] which is faster but with results slightly less accurate.

First we focus on A_0 (level 2) and compute a Taylor series for A_0 from interaction from those $4^3-3^3=37$ far cells A as

$$V_{A_0}^T(\mathbf{r})=\sum_A V_{\text{A}}^{\text{pole}}(\mathbf{r}-\mathbf{r}_A). \tag{5a}$$

Next we focus on A_0's child B_0 and calculate the Taylor series for B_0. It is computed as

$$V_{B_0}^T(\mathbf{r})=V_{A_0}^T(\mathbf{r}+\mathbf{r}_0)+\sum_B V_B^{\text{pole}}(\mathbf{r}-\mathbf{r}_B), \tag{5b}$$

where the sum of Taylor coefficients is over $8\times 27-27=189$ B cells, whose parents are nearest neighbors of A_0 but themselves are not nearest neighbors to B_0. Note that the far field for A_0, $V_{A_0}(r)$, is shifted by $\mathbf{r}_0$ to the center of B_0. Thus, $V_{B_0}^T$ represents the fields due to all atoms in cells labeled A or B in Fig. 2(c). This procedure is then repeated

TABLE I. Accuracy and speed of the cell multipole method on a dendrimer system (see caption for Fig. 1). A list of number of atoms N, deepest level L, number of cells M in the deepest level, average number of atoms per cell n, total CPU time (seconds) of the exact method Te, and of the CMM method Tc, total energy E (kcal/mol) and error, rms force error (kcal/mol·Å) and equivalent percentage errors. The timings of exact method for 199 329 atoms or more are extrapolated from the $N=15\ 333$ case. For each system, the second order results are given in the first line and the third order results in the second line.

N	L	M	n	Te	Tc	E	(δE)	rms δF (%)
453	3	512	0.9	0.95	0.51	−11 639.0	(−1.1)	0.200(0.43%)
					1.64		(0.08)	0.093(0.20%)
933	3	512	1.8	4.12	1.82	−24 004.6	(1.4)	0.169(0.37%)
					2.99		(−0.09)	0.087(0.19%)
1 893	3	512	3.7	18.6	3.44	−48 167.3	(−10.9)	0.196(0.43%)
					5.62		(−2.7)	0.096(0.21%)
3 813	3	512	7.4	69.8	6.6	−96 506.6	(−7.4)	0.171(0.37%)
					13.9		(−5.4)	0.081(0.18%)
7 653	4	4 096	1.9	287.0	14.8	−193 452.3	(−9.7)	0.364(0.69%)
					31.3		(−4.4)	0.180(0.34%)
15 333	4	4 096	3.7	1 193.0	38.8	−409 301.6	(81.7)	0.403(0.23%)
					59.8		(−10.6)	0.199(0.11%)
199 329	5	32 768	6.1	202 000.0	362.0	−5 321 043.0		
291 327	6	262 144	1.1	431 000.0	755.0	−7 777 442.0		
659 319	6	262 144	2.5	2 206 000.0	1719.0	−1 7602 242.0		
843 315	6	262 144	3.2	3 609 000.0	2203.0	−22 515 255.0		
1 211 307	7	2 097 152	0.58	7 446 000.0	3143.0	−32 340 180.0		

at level four for C_0 to obtain the Taylor series for C_0. Starting with level 2 cells A using Eq. (5a), all other deeper level far cells to C_0 are computed in the series of sums, Eq. (5b), each over a 189-cell domain with cell sizes which reduce gradually as the distance to C_0 becomes closer. During this process, cell size to distance ratio remains approximately a constant $\lesssim 1/2$.

Although the relation between a cell and the eight subcells leads to a tree structure, we find it more useful and efficient to use a Cartesian integer system to index cells at each level. This index system alleviates the use of recursive subroutine calls in the computer program (written in FORTRAN), and the calculation easily proceeds cell after cell at each level, and then, level after level. The Cartesian indexing also greatly facilitates algorithm improvements and verification.

IV. ACCURACY AND SPEED

The accuracy and speed of the CMM algorithm are assessed by calculations[10] on realistic polymers/dendrimers with up to 1.2 million atoms. A typical dynamics/minimization involves calculations of the forces and total energy, so the goal is to calculate them accurately and efficiently. The accuracy of the method is controlled by the orders in the multipole series [cf. Eq. (2)] and in the Taylor series [cf. Eq. (4)]. Both theoretical considerations and explicit tests indicate that the two series should be calculated in the same order of accuracy. We implemented the algorithm through the third order (the octopole level in the multipoles and third order in the Taylor series.) In Table I, we list the results obtained by applying CMM to a starburst dendrimer system[3] (from 453 to 1.2 million atoms) for the Coulomb interaction. At second order (the quadrupole level), the error in total energy is 0.02% and the rms error in forces is 0.4%. At octopole level, the errors are 0.009% for energy and 0.18% for force. This is consistent with the expectation from the theory that the errors are reduced by a factor $\approx d/R \lesssim 1/2$ for each additional order. For the London dispersions $1/r^6$, the errors in both energy and forces are extremely small (2×10^{-6} and 6×10^{-7} at second order). Similar results are also obtained on the clusters of poly(vinylidene fluoride) with up to one million atoms. Clearly, the accuracy at the quadrupole level is quite adequate, and thus we report all results at this level, unless otherwise stated.

Currently, the most common approach for treating large molecules is to cut off the interactions at a cutoff distance ($\simeq 10.5$ Å) and the potential is smoothed out by a cubic spline function over a range of 1 Å wide.[11] This leads to large errors.[12] For example, for the polymer system [a cluster of poly(vinylidene fluoride) with 1792 atoms, see Fig. 3 and caption there], the spline cutoff method leads to an rms force error of 43.8 [units are (kcal/mol)/Å], while CMM leads to 0.411 (0.218 at third order), two orders of magnitude smaller. The energy error is worse. These large errors due to the long-range tails ignored in the cutoff method can have profound effects on thermodynamical properties[13] and can lead to large errors in structures. Thus in Fig. 3, we compare the minimized structures for the polymer. The cutoff method gives a visibly distorted structure (rms deviation 0.812 Å), whereas, CMM produces a structure almost identical to the exact calculation (rms deviation 0.061 Å).

The total CPU time per atom, T/N, is plotted in Fig. 1. (All calculations are carried out on a Silicon Graphics workstation, SGI 380 one processor. The speed is about 5 MFLOPS.) Clearly, for the CMM method, this approaches a constant of about 2.6 msec per atom. For the exact method, T/N grows linearly with N. An extrapolation from the $N=15\ 333$ case leads to $T=2068$ h for the

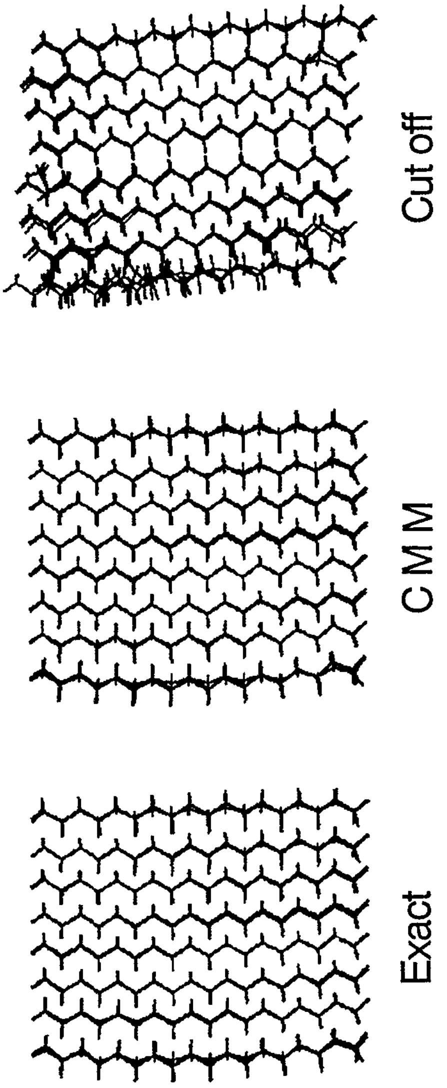

FIG. 3. Optimized structure for a 1792-atom poly(vinylidene fluoride) polymer (PVF_2). [PVF_2 has ($-CH_2-CF_2-$) chains packed together and exhibits at least four stable crystalline forms of which three are piezoelectric. Starting with form I we constructed clusters as multiples of the crystallographic unit cell, adding an H or F at the end of the chain to make $-CH_3$ or $-CF_3$, leading to a total of 1792 atoms with a size of $24.2\times34.2\times21.4$ Å. Progressively larger systems are then constructed.] There are eight layers of chains most of which are in the shadow of the top layer. The spline cutoff calculations lead to substantial structure deviations.

1.2 million atom system, which is far too slow in any standards. In contrast, the CMM can easily compute this 1.2 million atom system on the workstation, it only takes 0.87 h (see Table I).

V. DISCUSSION AND CONCLUSION

Some of the ideas developed in this method could have applications in other simulations. For example, the use of charge, dipole, and quadrupole, etc., as a representation for a collection of atoms could be very fruitful, as shown in Sec. II. This idea could be used more naturally in a protein simulation, where the charge of a peptide unit is often zero so the dominant $1/r$ interactions is already suppressed. Another example is to use a polynomial (Taylor) series to represent small local variations of the fields generated by distant sources.

The CMM algorithm developed here is quite suitable for the advanced computer architectures. The structure of the algorithm allows most parts to be easily vectorized for a vector or pipelined computer. In addition, CMM is simple to adapt to parallel computers, when the processors are connected as a $4\times4\times4$ (or $8\times8\times8$, etc.) grid. The three-dimensional space is naturally mapped into the processors and the deeper levels work independently. Because the computations are dominated by the deeper level cells and by pairwise interactions which are well localized in near-neighbor processors, we expect high efficiency. An implementation of this algorithm on a parallel computer is underway.

In conclusion, the cell multipole method is an efficient and practical approach to handle long range interactions, as demonstrated with calculations of systems having up to a million atoms. The workstation used here is well short of the state-of-the-art supercomputers. The vectorization and parallelization capability intrinsic to CMM should allow a very significant further speedup. Thus we believe that CMM will allow atomic level simulations on million atom systems to become quite practical.

ACKNOWLEDGMENTS

This research was partially funded by the Air Force Office of Scientific Research (AFOSR-88-0051) and by an EMRA from BP America (J. Burrington). The facilities at the MSC/BI has been supported by grants from DOE-ECUT, NSF-DMR-MRG, NSF-CHE, ONR, and from Allied-Signal Corp., Asahi Chemical, Asahi Glass, Xerox Corp., and Beckman Institute. This is contribution No. 8463 from the Division of Chemistry and Chemical Engineering at Caltech.

[1] W. C. Swope and H. C. Andersen, Phys. Rev. B **41**, 7042 (1990).

[2] P. E. Smith and B. M. Pettitt, J. Am. Chem. Soc. **113**, 6029 (1991).

[3] D. A. Tomalia, A. M. Naylor, and W. A. Goddard, Angew. Chem. Int. Ed. Engl. **29**, 138 (1990).

[4] Special issue ed.: M. L. Klein and P. J. Rossky, Chem. Phys. **129**, 163 (1989).

[5] A. W. Appel, SIAM J. Sci. Stat. Comput. **6**, 85 (1985).

[6] J. E. Barnes and P. Hut, Nature **324**, 446 (1986).

[7] L. Greengard and V. I. Rokhlin, J. Comput. Phys. **73**, 325 (1987).

[8] F. Zhao, Master thesis, MIT Artificial Intelligence Lab Report 995, 1987.

[9] The series is derived by writing $|\mathbf{R}-\mathbf{r}|^{-p}=R^{-p}[1+(-2\mathbf{r}\cdot\mathbf{R}/R^2+r^2/R^2)]^{-p/2}$ and treating the terms in the parentheses as a small quantity. The multipoles are obtained by writing quantities such as $(\mathbf{r}\cdot\mathbf{R})^2$ as scalar products of tensors and then symmetrizing the tensors with respect to x, y, z components.

[10] In force fields descriptions of polymers/dendrimers, besides the nonbond Coulomb interactions considered here (which typically accounts for more than 90% of total calculations), there are bonding interactions to describe stretching, bending, and torsions, and nonbonding Leonard-Jones 6–12 potentials, the hydrogen-bonding 10–12 potentials, etc. The system is equilibrated with the full force field. The time and accuracy of the calculations reported here refers only to the energy and force of the nonbonding Coulombic part for one equilibrated configuration (we tested several different configurations, the changes in results are negligible). In the energy minimization of the structure, the full force field is

used, with the nonbonding Coulombic part calculated either by the CMM or the cutoff methods.

[11] P. Weiner and P. Kollman, J. Comp. Chem. **2**, 287 (1981); B. R. Brooks *et al.*, *ibid.* **4**, 187 (1983).

[12] For more details, see H.-Q. Ding, N. Karasawa, and W. A. Goddard, Chem. Phys. Lett. **193**, 197 (1992).

[13] C. L. Brooks, B. M. Pettitt, and M. Karplus, J. Chem. Phys. **83**, 5897 (1985); C. L. Brooks, *ibid.* **86**, 5156 (1987).