FAMUSAMM: An Algorithm for Rapid Evaluation of Electrostatic Interactions in Molecular Dynamics Simulations

M. Eichinger, H. Grubmüller, H. Heller, and P. Tavan

Institut für Medizinische Optik, Theoretische Biophysik,
Ludwig Maximilians Universität München,
Theresienstr. 37, D-80333 München, Germany

e-mail: markus.eichinger@imo.physik.uni-muenchen.de

Key words:
molecular dynamics, protein dynamics, fast electrostatics computation, fast multipole method, multiple time step method,

Abstract

Within molecular dynamics (MD) simulations of protein-solvent systems the exact evaluation of long-range Coulomb interactions is computationally demanding and becomes prohibitive for large systems. Conventional truncation methods circumvent that computational problem, but are hampered by serious artifacts concerning structure and dynamics of the simulated systems. To avoid these artifacts we have developed an efficient and yet sufficiently accurate approximation scheme which combines the structure-adapted multipole method (SAMM) [Niedermeier and Tavan, *J. Chem. Phys*, **101**, 734-748] with a multiple-time-step method. The computational effort for MD simulations required within our fast multiple-time-step structure-adapted multipole method (FAMUSAMM) scales linearly with the number of particles. For a system with $36,000$ atoms we achieve a computational speed-up by a factor of 60 as compared to the exact evaluation of the Coulomb forces. Extended test simulations show that the applied approximations do not seriously affect structural or dynamical properties of the simulated systems.

Introduction

In many cases dynamical processes essentially contribute to protein function, e.g., in ligand binding or in enzymatic reactions. Representing a microscopic description of protein dynamics in the framework of classical mechanics, molecular dynamics (MD) simulations[1–3] can serve as a tool to interpret experimental data and to provide predictions on hitherto unobserved processes.

MD simulations of protein dynamics pose a computational challenge, since suitable simulation systems have to include the native environment (e.g., water and lipid molecules) of these biological macromolecules[4–7] and, therefore, typically have to comprise several $10,000$ atoms. Furthermore, they have to use femtosecond integration time steps to enable smooth descriptions of the fastest degrees of freedom. Simulations of such systems are currently limited to nanoseconds even if the most powerful supercomputers are employed. Although there are biochemically important processes which actually occur on a nanosecond time scale[8] and have been successfully described by MD simulations,[9,10] most biochemical processes occur on time scales well above nanoseconds and, thus, are inaccessible to conventional MD methods.

In the MD approach the classical Newtonian equations are solved numerically to obtain the motion of all atoms.[11–13] For each numerical integration step the Coulomb sum

$$U \propto \sum_{i}^{N} \sum_{j<i} \frac{q_i q_j}{|\mathbf{r}_i - \mathbf{r}_j|} \tag{1}$$

between all pairs of atoms (i, j) with partial charges q_i and positions $\mathbf{r}_i$ has to be evaluated. In common force fields for protein and solvent molecules most atoms carry partial charges; therefore the evaluation of U dominates the computational effort in MD simulations as it scales quadratically with the number N of charged particles.

In truncation methods,[14] which simply neglect long range electrostatic interactions beyond a cutoff distance of typically $8 - 10$ Å, the computation of U scales with N instead of N^2 and, consequently, such methods are widely used in MD simulations. However, the truncation of electrostatic interactions leads to serious errors in forces, energies and other observables. These errors are generally too large to be acceptable for descriptions of protein dynamics.[15–18] For instance, using a cutoff distance of 10 Å the relative errors in the computation of electrostatic forces are of the order of 10%.[19] One of the major artifacts concerning the description of protein dynamics, that is entailed by these sizable errors, is a suppression of dynamical correlations between slowly fluctuating protein modes, as these are mainly coupled by the neglected long-range Coulomb forces.[16,18] Thus, truncation of electrostatics trades a description of protein dynamics which is computationally prohibitive but adequate in the sense of basic physics for a description which is computationally manageable but physically inadequate*.

As a result of the above considerations, one of the major tasks which has to be tackled in the strive for improved MD methods is the design of approximate algorithms for computationally efficient and sufficiently accurate evaluation of long-range Coulomb interactions and integration of the Newtonian equations.

However, any attempt to solve the problems stated above is confronted with two difficulties: First, requiring efficiency and accuracy on equal footings typically represents a contradiction, since approximate algorithmic schemes tend to gain computational efficiency at the cost of accuracy; that contradiction already has shown up in the above discussion of truncation methods and has to be taken care of in the design of any new approximation scheme. Second, it is much simpler to estimate the computational complexity of an approximate MD algorithm, since this is just a matter of counting floating point operations, than to judge its accuracy. In fact, as discussed in refs. 16 and 20, the latter task is highly nontrivial due to the complicated, nonlinear, and chaotic dynamics of macromolecular systems and requires careful comparisons of a variety of statistical observables on structural and dynamical properties which have to be extracted from extensive sample simulations. For instance, the particular artifact of the truncation method noted above, i.e., the suppression of dynamical correlations, became apparent only after the progress of computer technology had enabled simulations of sufficiently long duration. To the extent that quality assessments of approximate MD algorithms will become necessary in the present paper, we will adopt the problem adapted approach developed in refs. 16 and 20 and will employ the procedures and some of the statistical observables which have been suggested there as useful and suitable tools to judge algorithmic accuracy (see also further below).

Up to now various approaches have been suggested which aim at a more efficient approximate treatment of the long-range electrostatic interactions in MD simulations. These approaches may be classified into two categories: multipole[21–26,18,19] and multiple-time-step methods.[27–33,16]

Multipole methods approximate the long-range forces originating from a group of point charges by truncated multipole expansions of their electrostatic potential. Using a hierarchy of grids for subdivision of space, nested on multiple scales, and a corresponding hierarchical organization of charge groups and multipole expansions[22] a computational

*Also the computation of van der Waals forces required in MD simulations scales with N^2; However, as these forces decay much more rapidly with increasing interaction distance than the Coulomb forces, they can safely be truncated at about 10 Å and, therefore, are computationally inexpensive .

complexity of $O(N log N)$ is achieved. By additionally using a hierarchy of local Taylor expansions for the evaluation of the electrostatic forces acting on particles Greengard and Rokhlin have constructed the so-called *fast multipole method* (FMM) that even scales with $O(N)$ for large systems.[23–25] Several implementations of this method suited for MD simulations have been reported.[34–36]

For MD simulations of biomolecules the FMM-type grouping of charges, defined by a fixed and regular subdivision of space, requires multipole expansions of rather high order as to achieve sufficient numerical accuracy.[24] If, instead, charge grouping is adapted to structural and dynamical properties of the simulated biomolecules, multipole expansions can be truncated at low orders without large sacrifices of accuracy.[19,26,18] Accordingly, such a structure adapted multipole method (SAMM) provides a further substantial speed-up for MD simulations.

Multiple-time-step methods are based on the observation that forces between distant atoms generally exhibit slower fluctuations than forces between close atoms. Therefore, without significant loss of accuracy, the more slowly fluctuating forces can be treated with longer integration step sizes. The required separation of forces can be implemented, e.g., by grouping atom pairs into distance classes. Simple multiple-time-step methods define only two distance classes, an inner and an outer one,[27] whereas more advanced methods employ a hierarchy of such classes.[28–33,16]

As one can expect from the general considerations presented further above, the enhanced efficiency of multipole and multiple-time-step methods is obtained at the cost of various kinds of algorithmic artifacts. A typical example is the artificial introduction of "algorithmic noise", which arises from discontinuous and discretized representations of continuous and smooth functions and processes. Such noise effectively can act like an uncontrolled stochastic force on to the atomic motion, and can turn the simulated dynamics into a Langevin-type dynamics with the nasty property, that the algorithmic noise is not properly balanced by a corresponding friction term and, therefore, constantly adds heat to the simulated system. By thorough studies of sample simulations these artifacts have been shown to be negligible in the case of the SAMM method[19,26,18] and to be optimally small in the case of a particular multiple-time-step procedure, the so-called DC-1d extrapolation scheme, which has been suggested and carefully compared with other methods in ref. 16.

From a general point of view, multipole methods and multiple-time-step methods exploit different regularities of Coulomb interactions as to obtain enhanced computational efficiency: the former make use of regularities *in space*, whereas the latter exploit regularities *in time*. This complementarity led us to propose[16] that a combination of both concepts should be capable to render additional speed-ups. In view of the proven superiority of the SAMM method as compared to fixed grid FMM procedures on the one hand, and of the DC-1d scheme as compared to other multiple-time-step schemes on the other hand, these two methods appear to be the natural candidates for the envisaged combination. It is the purpose of the present paper to show how a suitable combination of these methods can be achieved and to demonstrate that the resulting *fast multiple-time-step structure-adapted multipole method* (FAMUSAMM) actually preserves and combines the advantages of the parent methods with respect to enhanced computational efficiency and to the lack of substantial algorithmic artifacts. In particular we will demonstrate by comparison of a combination of SAMM with other multiple-time-step schemes, that the DC-1d procedure actually is the method of choice.

Windemuth[37] has recently presented a first step towards a combination of multipole and time-step methods; using a simple variant of a multiple-time-step method with only two distance classes and a conventional FMM algorithm he achieves considerable speed-ups. Unfortunately, a quality assessment is lacking in Windemuth's paper. Zhou and Berne[38] have reported a more advanced combination of the multiple-time-step algorithm RESPA[31] with a slightly modified fast multipole method using a rectangular subdivision of the simulation volume. Here, quality assessments are based on short MD simulations in the ps range. For MD simulations with periodic boundaries[39–41] a fast method based on a combination of RESPA and the smooth particle mesh Ewald method[42] has been described.[43] The latter method is particularly useful for simulations of crystalline structures, e.g., protein crystals. Also, for the simulation of non-crystalline systems, like in lipid membranes soluted proteins,[4] periodic boundaries have been used as an approximation due to the availability of efficient algorithms. In the latter case, however, it is *a priori* not clear whether that approximation is justified (for a discussion of possible artifacts see, e.g., ref. 44–47). In our paper we develop an efficient algorithm to speed up simulations of non-periodic systems, in which, typically, the environment is effectively included through the use of appropriate boundary forces (see, e.g., refs 44,48). We provide a careful analysis of possible algorithmic artifacts and their influence on structural and dynamical properties of a test system.

We intend to achieve high efficiency by combining SAMM and a multiple-time-step scheme. To that aim, we first outline the basic features of the parent methods and, subsequently, present our new algorithm FAMUSAMM. As will be explained in the following sections, in our multipole method charge grouping will be based on the specific structural features of macromolecules rather than on fixed subdivisions of space as used in the methods cited above. In particular, our combination of SAMM with a multiple-time-step method is expected to be more efficient and less memory consuming, since we intend to apply the multiple-time-step scheme to local Taylor expansions of the electrostatic potential rather than to Coulomb forces acting on individual atoms. Having defined suitable methods to evaluate FAMUSAMM's accuracy, we present the results of our accuracy assessment. After documenting the enhanced computational performance of our method, we discuss and summarize our results.

The FAMUSAMM algorithm

As indicated above we intend to combine two well-established methods for rapid evaluation of long range Coulomb interactions into a new and improved algorithm. To enable an understanding of the computational strategy to be implemented by our combination, we first sketch the basic concepts of its parent methods, SAMM and the multiple-time-step method DC-1d.

Structure Adapted Multipole Method

The SAMM method[19, 18, 26] exploits structural features of biomolecules to define a hierarchical grouping of partially charged atoms into local charge distributions, whose multipoles are subsequently used for evaluation of electrostatic interactions at large distances.

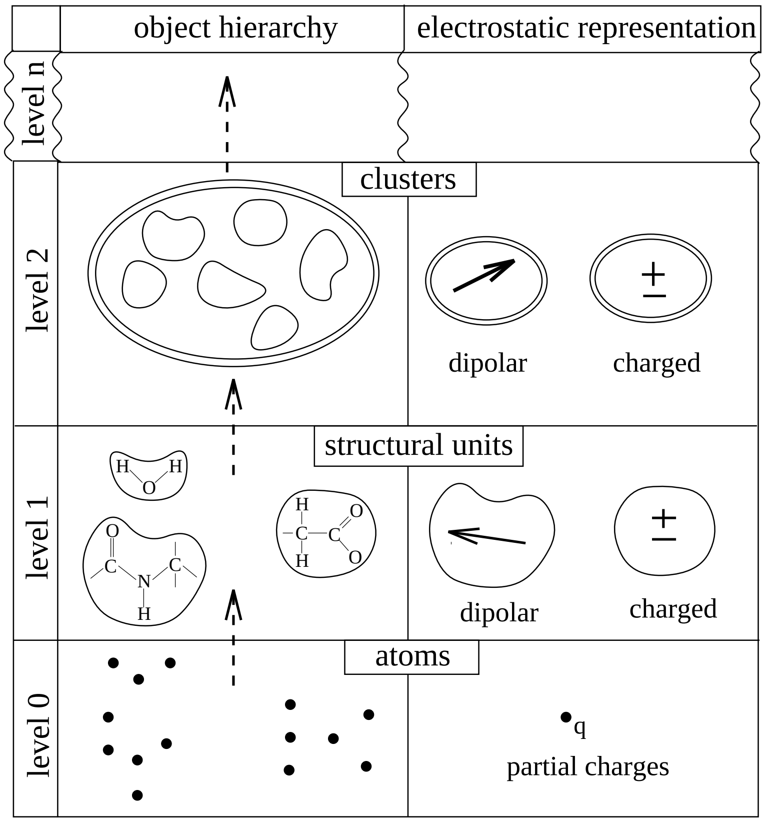

Figure 1: Structure adapted hierarchical description of biological macromolecules. Filled circles represent atoms, structural units are surrounded by a single-line border, and clusters are surrounded by a double-line border.

Figure 1 illustrates the hierarchy of charge distributions ("objects") obtained by the SAMM-type grouping (left) and the electrostatic representations chosen for the resulting objects (right). Both, the definition of the objects and the choice of their electrostatic representations rest on the observation, that covalently bound atoms within biomolecules intrinsically form dynamically stable local groups, which either carry integer elementary charges or are uncharged, but dipolar. In case that the latter dipoles are small, the local groups are called non-polar and their electrostatics may be neglected.[19,18]

This observation suggests, that at larger distances the lowest multipole moments of local groups can provide an accurate description of the electrostatics within biomolecules. Thus, charged local groups should be described by monopoles and dipolar groups by dipoles for the description of their electrostatic interactions at large distances. In the SAMM method that concept represents the guideline for the definition of the object hierarchy illustrated in Figure 1. As shown in the figure, at the bottom of the object hierarchy (level 0) no local groups are formed; instead individual atoms (filled circles) are considered as the basic objects and are electrostatically represented by their partial charges†. At

†In a heteronuclear molecule each atom carries a partial charge; however, usual parameteriza-

small distances, i.e., at distances smaller than a predefined distance D_1, their interactions are evaluated by means of the Coulomb sum, eq. (1). For a description of electrostatic interactions at larger distances covalently bound atoms are grouped into *structural units* which are compact, typically include three to ten atoms, and form the first level of the object hierarchy (level 1). According to their respective electrostatic properties, SAMM distinguishes three types of structural units: "neutral units" consist of uncharged atoms only; they need not be considered and therefore are omitted in Figure 1. "Dipolar units", e.g., water molecules or peptide groups of a protein backbone, are composed of partially charged atoms, exhibit a vanishing net charge, and are represented by a dipole (arrow). "Charged units", e.g., charged side groups of amino acids, carry an integer net charge and are represented by a monopole. At all higher levels (level 2 and further up) pertaining to interactions at still larger distances, i.e., at distances larger than a second, predefined interaction distance D_2, objects of the preceding lower level are grouped into *clusters* by means of self-organizing and adaptive vector quantization techniques.[49] For a predefined number of clusters these techniques manage to minimize cluster extension and, thereby, also the approximation error caused by the truncation of the multipole expansions.[19,18] Depending on their total charge the clusters are represented either by dipoles or by monopoles[‡]. We use $D_1 = 10\,\text{Å}$ and $D_2 = 16\,\text{Å}$ for the interaction distances associated with the hierarchy levels; that choice has been demonstrated to guarantee highly accurate descriptions.[19]

Computationally the hierarchical multipole expansions of SAMM are defined and evaluated by a *bottom-up* scheme: The monopoles and dipoles of the structural units at level 1 are calculated from the individual charges of their enclosed atoms; at the second level the monopoles and dipoles of the clusters are obtained from the respective multipole moments of their embodied structural units and so forth; for further details see ref. 18.

Note that conformational changes in the protein or diffusive motions of solvent molecules would lead to an increase of cluster sizes in the course of an MD simulation, if the grouping of structural units into clusters would be chosen fixed. As a result, the accuracy of the electrostatic representation by the truncated multipole expansions would decrease. To maintain the accuracy of description SAMM periodically regroups the structural units into new clusters applying the adaptive techniques mentioned above. In principle, such regrouping represents a discontinuity of description and, thus, is a source of algorithmic noise. However, the chosen adaptive reclustering procedure assures, that the redistribution of structural units among clusters is kept minimal, and, therefore, also minimizes algorithmic noise.[19,18]

In addition to the diligent choice of structural units and of optimally compact clusters the SAMM method provides a further means to reduce the approximation error connected with the truncated multipole expansions, i.e., the choice of optimal reference points of the multipole expansions. This strategy is based on the fact that the first non-vanishing moment of a multipole expansion, in contrast to the higher moments, is independent of the reference point. As SAMM considers only the first non-vanishing moment, the truncation

tions of protein force fields are chosen such that small partial charges are neglected while the sum of partial charges is locally kept at integer values.

[‡]In contrast to the original SAMM version, we now also combine monopoles of opposite sign into dipolar objects at higher levels, if the total charge of these objects vanishes; this procedure allows to represent the electrostatics of neighboring ion pairs at large distances as dipoles.

error can be made optimally small by an adequate choice of the reference point.[19,18]

Whereas the original version of SAMM[19] solely comprises a hierarchical multipole scheme, its enhanced version[26] also utilizes FMM-strategies.[23,24] Here, local Taylor expansions of the electrostatic potential originating from distant charge distributions and their associated multipole moments are used to speed up the calculation of forces acting on partial charges in a given neighborhood. A careful analysis of sample simulations has shown that the local Taylor expansion may be safely truncated at second order, if the SAMM specific distance classes are applied, i.e., if forces originating from objects more distant than D_1 or D_2 are considered.[18,26] Thus, the fast version of SAMM approximates the corresponding electrostatic potential $\Phi(\mathbf{R})$ at position $\mathbf{R} = \mathbf{R}_\mathrm{o} + \mathbf{r}$, within the volume of a structural unit or cluster in terms of its value and derivatives at the optimized reference point $\mathbf{R}_\mathrm{o}$ of that object by the local Taylor expansion

$$\Phi^L(\mathbf{R}_\mathrm{o} + \mathbf{r}) \quad = \quad \Phi_\mathrm{o} + \sum_{\alpha=1}^{3} K_\alpha r_\alpha + \frac{1}{2} \sum_{\alpha,\beta=1}^{3} T_{\alpha\beta} r_\alpha r_\beta \,, \tag{2}$$

where the expansion coefficients are defined by

$$\Phi_\mathrm{o} \equiv \Phi(\mathbf{R}_\mathrm{o}), \quad K_\alpha \equiv \left.\frac{\partial \Phi(\mathbf{R})}{\partial R_\alpha}\right|_{\mathbf{R}=\mathbf{R}_\mathrm{o}} \quad \text{and} \quad T_{\alpha\beta} \equiv \left.\frac{\partial^2 \Phi(\mathbf{R})}{\partial R_\alpha \partial R_\beta}\right|_{\mathbf{R}=\mathbf{R}_\mathrm{o}} \,. \tag{3}$$

In eqs. (2) and (3) Greek indices enumerate Cartesian coordinates.

Figure 2 illustrates the FMM aspect of SAMM for the lowest three hierarchy levels, $H = 0, 1, 2$. As a function of object distance d it is shown how objects at different hierarchy levels contribute (long arrows) to the electrostatic force acting on a selected atom (⌘). Three corresponding distance classes ($d \leq D_1, D_1 \leq d \leq D_2, D_2 \leq d$) are separated by vertical dashed lines. Within these distance classes the figure displays those objects, which contribute to the force acting on objects of the respective hierarchy level. These objects, which contain the selected atom and which we correspondingly call selected objects, are shown within the leftmost column. For the evaluation of the total electrostatic force on the selected atom the following algorithmic steps have to be performed in a *top-down* fashion:

Step 1: At the highest hierarchy level ($H = 2$ in Figure 2) the electrostatic potential generated by the multipole moments of all those clusters, which are separated from the selected cluster by more than the second interaction distance D_2, is approximated within the selected cluster by a local Taylor expansion of the electrostatic potential using the reference point (▲).

Step 2: Now the local Taylor expansion of the selected cluster is transformed (short arrows) to local Taylor expansions of its embodied structural units. This procedure is called "inheritation" and is the key to the highly efficiency of FMM schemes. The transformation is accomplished by shifting the reference point from the center of the selected cluster (▲) to the centers (■) of its embodied structural units.[18,26] The resulting local Taylor expansions of the structural units now describe the same electrostatic potential as their "parent" local Taylor expansion.

Step 3: At the next lower hierarchy level ($H = 1$ in Figure 2) we consider additional

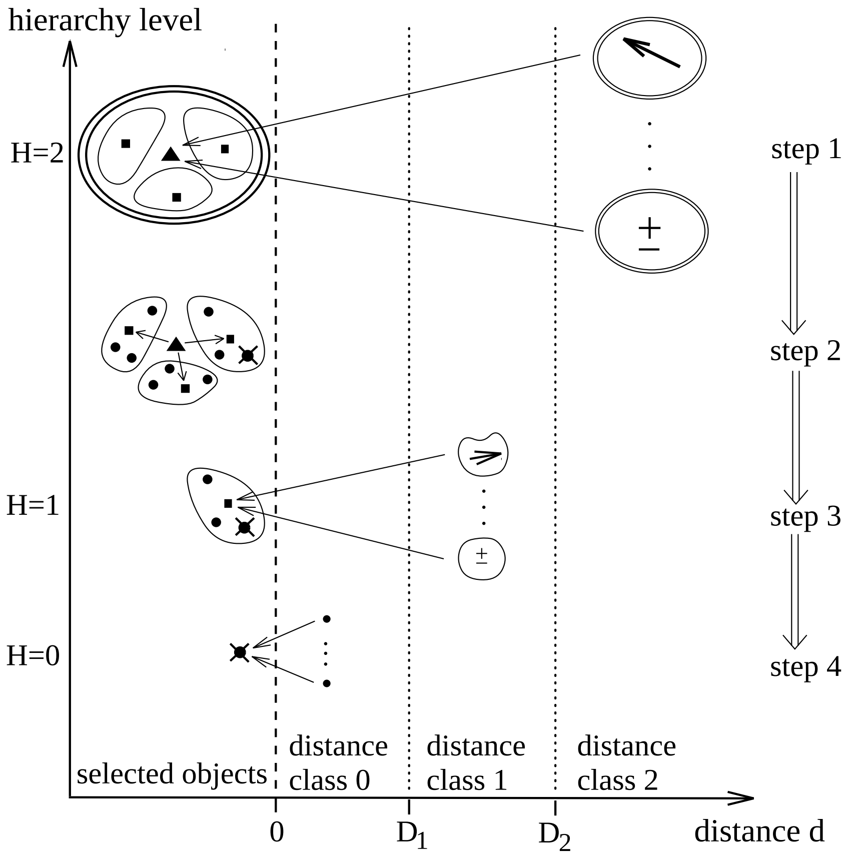

Figure 2: Illustration of algorithmic steps connected with the local Taylor expansion in the force calculation of the fast version of SAMM; the contributions to the force on a selected atom (✖) originating from objects located within various distance ranges and associated to corresponding hierarchy levels H are indicated by long arrows; the reference points of the local Taylor expansions are denoted by triangles (▲) for clusters ($H = 2$) and by squares (■) for structural units ($H = 1$); short arrows starting at a reference point indicate the "inheritation of local Taylor expansions"; for explanation see text.

electrostatic interactions, i.e., those of the selected structural unit with structural units in a distance range between D_1 and D_2. Their contribution to the electrostatic potential within the selected structural unit is again approximated by a local Taylor expansion and is added to the local Taylor expansion inherited from the higher levels. The resulting local Taylor expansion approximates the electrostatic potential generated by all atoms more distant than D_1.

Step 4: At the lowest hierarchy level, $H = 0$, the local Taylor expansion of the selected structural unit is evaluated at the position of the selected atom. Interactions with atoms at distances smaller than D_1 are calculated directly from the corresponding Coulomb sum, eq. (1), and are added.

The above presentation of the FMM-strategy utilized by SAMM completes the outline of that algorithm. We have seen how SAMM takes advantage of structural properties of protein-solvent systems and how this method exploits distance dependent *spatial* regularities of the corresponding electrostatic potential to construct a computationally fast and yet accurate approximation scheme.

Multiple Time Step Methods

Another class of approximation schemes, the multiple-time-step methods[28, 29, 32, 16, 33] , exploit complementary properties of protein electrostatics for the same purpose; they take advantage of *temporal* regularities. The so-called distance class methods, in particular, are based on the observation that forces originating from distant atoms fluctuate more slowly than forces from atoms nearby (see Figure 3). The slowly fluctuating forces may be evaluated less frequently than the fast ones and may be extrapolated at the time steps in between. Such extrapolation is required as the numerical integration of the dynamical equations needs all forces at every integration time step $\tau = 0, 1, 2, ...$, which discretizes the simulation time $t = \tau\Delta t$, where Δt is the integration time step size[§].

The left part of Figure 3 illustrates how distance classes j can be defined by a set of increasing radii R_{j+1}. The set of atoms $\{m\}$ at positions $\mathbf{r}_m$ satisfying $R_j \leq |\mathbf{r}_l - \mathbf{r}_m| < R_{j+1}$ makes up the distance class j of particle l at position $\mathbf{r}_l$. As is also indicated in Figure 3 (right part), for each particle l the sum of forces $\mathbf{F}^{(j)}$ arising from particles in distance class j is calculated explicitly every n_j-th integration time step (filled squares). Each of these time steps is called a macro integration step. A common choice[16] is $n_j = 2^j$. Correspondingly, the $2^j - 1$ elementary time steps within the cycle of a macro integration step are called micro integration steps. As sketched in Figure 3, at each step $\mathbf{F}^{(j)}$ is estimated from two forces calculated at previous macro integration steps by

$$\mathbf{F}^{(j)}_{n_j k+i} = a^{(j)}_i \mathbf{F}^{(j)}_{n_j k} + b^{(j)}_i \mathbf{F}^{(j)}_{n_j (k-1)} \tag{4}$$

using appropriate extrapolation coefficients $a^{(j)}_i$ and $b^{(j)}_i$. The lower index of $\mathbf{F}$ denotes the absolute integration time step number and is expressed in terms of the macro integration step $k = 0, 1, 2, \ldots$ and the cyclic micro integration step $i = 0, \ldots, n_j - 1$; $\mathbf{F}^{(j)}_{n_j k}$ and $\mathbf{F}^{(j)}_{n_j(k-1)}$ are explicitly calculated at the macro integration steps k and $k - 1$.

The hierarchical extrapolation procedure sketched above is capable to save an enormous amount of computer time as it frequently avoids the most time consuming step, i.e., the exact evaluation of all interactions. Here computational speed is gained at the cost of an increased demand for memory: for each atom and each distance class two previous forces have to be kept in memory.

Various choices for the extrapolation coefficients a^j_i and b^j_i have been discussed;[16, 33, 37] both the linear extrapolation (illustrated in Figure 3) and the so-called DC-1d algorithm

[§]In MD simulations usually Δt is set to 1 fs, which is short enough to smoothly describe the fastest degrees of freedom in protein simulations.

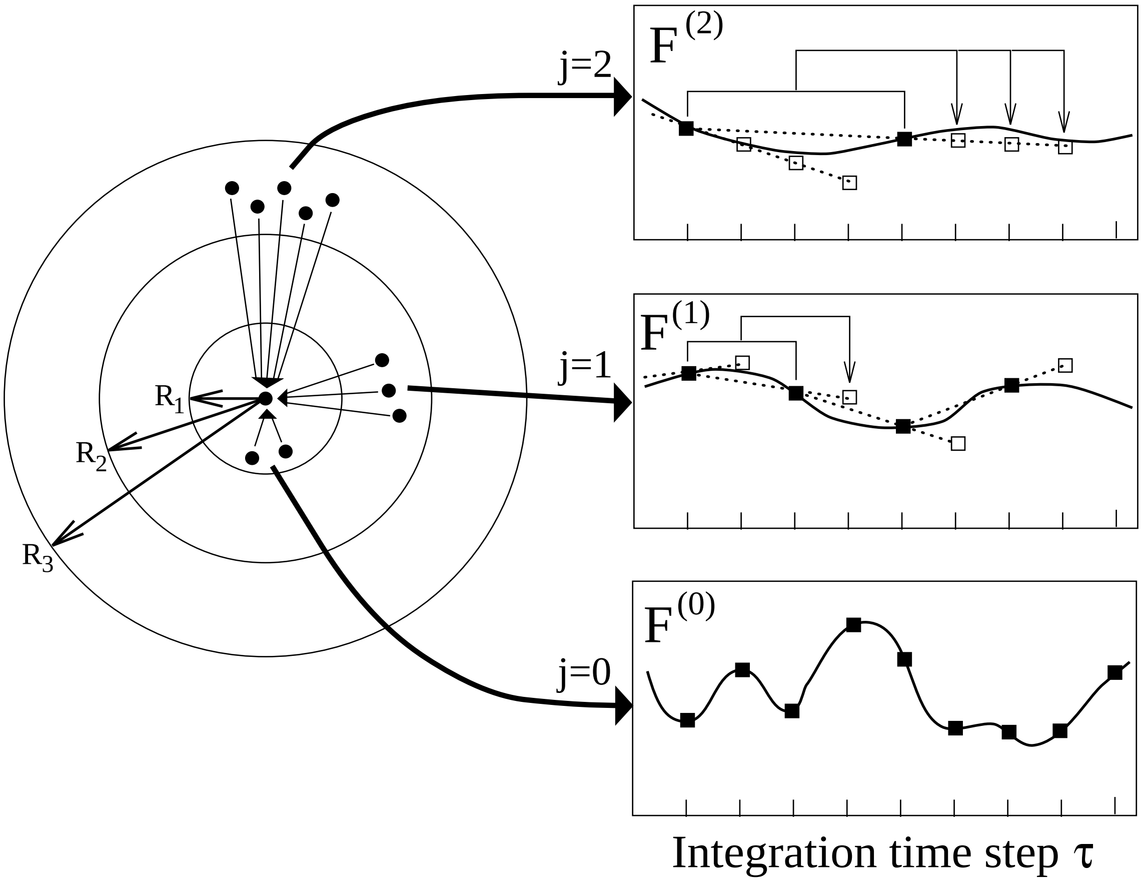

Figure 3: (left) Distance classes $j = 0, 1, 2, ..$, are defined for an atom (central dot) by a set of radii R_{j+1} generating a series of spherical shells; (right) also shown is for each distance class the temporal evolution of the total force $\mathbf{F}^{(j)}$ acting on the selected atom originating from all atoms in the respective distance class; along with the exact forces (solid line) and their exact values (filled squares) also linear force extrapolations (dotted lines) and resulting force estimates (empty squares) are depicted.

have been found promising.[16] Although the linear extrapolation defined by

$$a_i^{(j)} = 1 + \frac{i}{n_j} \qquad \text{and} \qquad b_i^{(j)} = 1 - a_i^{(j)} \, . \tag{5}$$

entails smaller discontinuities for the extrapolated forces than the DC-1d algorithm, it leads to a larger energy transfer into a simulation system by algorithmic noise.[16] The DC-1d scheme employs the coefficients

$$a_i^{(j)} = \frac{3n_j^2 - 2n_j + 1}{n_j(n_j+1)} - 3i\frac{n_j - 1}{n_j(n_j+1)} \qquad \text{and} \qquad b_i^{(j)} = 1 - a_i^{(j)} \quad . \tag{6}$$

It is a priori not clear, whether the quoted properties of these extrapolation schemes will pertain, if they are combined with the SAMM procedure into our new FAMUSAMM scheme, which will be explained in the next section. Therefore, using test simulations we will subsequently check both extrapolation methods for their suitability within our combination method.

Combination of SAMM with a multiple-time-step method

Before starting with the detailed outline of our FAMUSAMM method we want to annotate on an important progress concerning its implementation as compared to that of SAMM: The first two SAMM versions[19,18,26] made use of only two hierarchy levels, which restricted their applicability to medium-sized systems comprising less than about $5,000$ atoms. To gain substantial speed-ups also for larger systems, we decided for the design of FAMUSAMM to implement a fast SAMM method with an arbitrary number of hierarchy levels. That number is now determined by the number of atoms in a given simulation system and is chosen automatically in the start-up phase of a simulation.[50]

a) Basic concepts

Our FAMUSAMM algorithm has been designed to further reduce the computational effort for the evaluation of Coulomb forces by exploiting regularities of these forces in space *and* time. The basis of its development has been the identification of the most time-consuming steps within the fast SAMM algorithm. Only upon such an analysis one may expect that a combination with a multiple-time-step method can render additional speed-ups: A detailed inquiry of fast SAMM has shown,[50] that the essential time consuming steps are (i) the calculation of contributions to the local Taylor expansions (see Figure 2, steps 1 and 3) and (ii) the calculation of the electrostatic forces for the innermost distance class (see Figure 2, step 4). Therefore, we have focused our efforts on these two steps.

b) Multiple-time-step extrapolation of local Taylor expansions

First consider step (i) and remember, that the hierarchy of interactions, which is employed for the evaluation of the local Taylor expansions and is depicted in Figure 2, is defined by a hierarchy of distances. This hierarchy of distances is intimately related to the particular concept of distance classes underlying the multiple-time-step schemes as discussed above. Therefore, an attempt to combine these methods seems natural:

In multiple-time-step methods the *total force* $\mathbf{F}^{(j)}$ acting on a selected atom originating from distance class j is extrapolated. Since we now want to use a multipole method, we have to identify a corresponding expression for this force. As we have seen in our outline of the SAMM algorithm, the total force originating from outer distance classes and acting on elements contained in a cluster or a structural unit is computed by means of a local Taylor expansion. Such a local Taylor expansion for an object on a given hierarchy level is — except for the topmost level — made up of two components: (a) a contribution inherited from the next higher hierarchy level (see Figure 2, step 2) comprising the electrostatic interactions with objects in all outer distance classes, and (b) contributions from objects within the same hierarchy level. From the latter one can calculate the desired total force $F^{(j)}$ and, therefore, we apply multiple-time-step extrapolations to these contributions. Here, instead of extrapolating the forces directly, we extrapolate the coefficients Φ_0, K_α and $T_{\alpha,\beta}$ of the local Taylor expansions, from which the forces derive[¶].

[¶]Note, that this strategy also saves computer memory, since the local Taylor expansion of a

c) Inner class force extrapolation

Because of the choice of 10 Å for the interaction distance D_1 rendering quite a few, typically about 400, interaction partners for a given partial charge within the innermost distance class ($H = 0$) of SAMM, the second essential time consuming step (ii) of that method is the calculation of electrostatic forces by the Coulomb sum, eq. (1). In fact, this step is equivalent to the one carried out in conventional truncation methods using a cut-off distance D_1. To reduce that effort by application of a multiple-time-step method, we split this SAMM distance class into two new distance classes $j = 0$ and $j = 1$ with the respective radii $R_1 = 5$ Å and $R_2 = 10$ Å. Within these new distance classes we extrapolate forces directly by a multiple-time-step scheme as described in the section before. Distance classes of similar size have been successfully used previously.[16]

Due to the introduction of these new distance classes, the expensive computation of about 400 interactions per particle is now performed only at *every second* integration time step, whereas in distance class $j = 0$ only a few interactions, about 50, have to be evaluated at *every* integration step. Hence, the overall computational effort for the innermost SAMM distance class is reduced nearly by a factor of two.

d) Interaction list updates

Generally, MD methods, which are based on distance classes, use interaction lists to eliminate the need for time consuming interaction partner searches at each integration time step.[32] In FAMUSAMM that strategy is adopted as follows: For each object at a particular hierarchy level (atom, structural unit, cluster) an associated interaction list points to all objects, which belong to the same hierarchy level and meet the distance criterion described above (cf. Figure 2). Because distances between atoms, structural units and clusters vary during an MD simulation, the interaction lists have to be updated from time to time and are likely to change upon these updates. As a result, the forces generated by changing numbers of objects within a given distance class will change discontinuously. The reclustering procedure of SAMM mentioned above causes similar discontinuities. However, the occurrence of such discontinuities disables the application of multiple-time-step extrapolation procedures, since the latter are based on strict smoothness assumptions: Only if the interaction lists and cluster compositions are kept fixed for a sufficiently large [$\gg 4$, see subsection f)] number of integration time steps, multiple-time-step extrapolations may be applied. Test simulations have shown, that for our FAMUSAMM interaction lists update periods covering up to 128 integration steps can be used; reclustering periods may be even longer and are chosen as integer multiples of the former periods.

e) Verification of local Taylor expansion smoothness

As described above, an essential new feature of FAMUSAMM as compared to previous multiple-time-step schemes is the extrapolation of local Taylor expansion coefficients for outer distance classes instead of the extrapolation of forces acting on individual atoms. To verify the validity of this approach it remains to be checked, whether the smoothness conditions required for multiple-time-step extrapolations actually hold for the local Taylor expansion coefficients. Extended test simulations have demonstrated, that this is the case, indeed, and that local Taylor expansion coefficients exhibit decreasing amplitudes and curvatures with increasing hierarchy level H.

---

selected object is used to calculate the forces on all particles contained in that object; storage of all these forces would inevitably consume much more memory.

f) Initialization

A prerequisite for the application of the multiple-time-step extrapolation at a given time step in a simulation is the accessibility to two previously explicitly calculated forces or local Taylor expansion coefficients. As these are absent in the start-up phase of a simulation or, due to the associated discontinuities, cannot be used after an update of the interaction lists, special precautions have to be taken for these periods.[32] To be specific, the extrapolation of forces or of local coefficients for distance class j requires their values at two previous macro integration steps which are separated by $2^j - 1$ basic integration steps (see Figure 3). To provide these values for the initialization periods FAMUSAMM performs exactly $T_j = 2^j$ conventional fast SAMM steps until all necessary forces (for $j = 1$) or local coefficients (for $j \geq 2$) are explicitly calculated. Obviously, the initialization periods, after which multiple-time-step extrapolations can be applied, have different durations T_j for the various distance classes. For the computationally most expensive distance classes, $j = 1$ and $j = 2$, the initialization lasts only 2 and 4 integration steps, respectively. From these numbers it is clear that the efficiency loss due to the initialization phases is negligible, if periods of 64 or more steps are used for the update of the interaction lists.

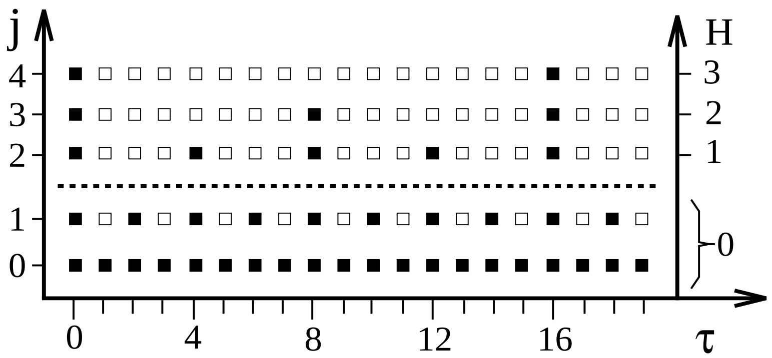

Figure 4: Extrapolation scheme of FAMUSAMM; τ denotes the integration step, H the SAMM hierarchy level and j the multiple-time-step distance class; filled squares mark the integration steps at which forces originating from objects in distance class j are explicitly calculated, whereas empty squares indicate computationally inexpensive extrapolations. The dashed lines separate the regime of force extrapolation in distance class $j = 1$ ($H = 0$) from the regime of local Taylor expansion extrapolation in the outer distance classes $j \geq 2$ ($H \geq 1$); forces originating from partial charges in the innermost distance class $j = 0$ are never extrapolated.

g) The FAMUSAMM algorithm

Figure 4 illustrates the multiple-time-step scheme of FAMUSAMM by which long range Coulomb interactions are evaluated. For a simulation period well after an initialization phase three hierarchy levels H and a sequence of 20 integration steps are shown. Filled squares indicate explicit fast SAMM force calculations, empty squares extrapolations. The dashed line separates the Coulomb sum interaction classes ($j = 0, 1$) from distance classes which interact by multipole expansions ($j \geq 2$). For class $j = 1$ the extrapolation of forces is used, i.e., the respective forces are evaluated only at every second integration step and are extrapolated otherwise; whereas, the electrostatic interactions of partially charged atoms closer than $R_1 = 5\,\text{Å}$ (see Figure 3, $j = 0$) are calculated at every integration step.

For distance classes $j \geq 2$, which correspond to hierarchy levels $H \geq 1$, the multiple-time-step scheme is applied to the local Taylor expansion coefficients. Here, the local Taylor expansion coefficients of lower hierarchy levels have to be recalculated more often than those of higher hierarchy levels, which, in turn, can be extrapolated more often. As suggested in a previous multiple-time-step method,[16] explicit recalculation is performed every 2^j-th integration step.

Appendix A summarizes the FAMUSAMM algorithm using a pseudo-code description. The algorithm has been implemented in the MD simulation program EGO_VIII[51] in a sequential and a parallelized version; the latter is suited for distributed memory parallel computers, e.g., IBM SP2, Cray T3D, Parsytec CC and workstation clusters under PVM or MPI. Details on the parallelization strategy will be given in a forthcoming publication.[52]

As is apparent from the large number of empty squares depicted in Figure 4 FAMUSAMM replaces quite a few explicit force or local Taylor expansion calculations by extrapolations for the sole purpose of computational efficiency. As noted already in the introduction, in MD-simulations the inaccuracies associated with these FAMUSAMM approximations may effectively act as sources of uncontrolled algorithmic noise. It is the aim of the remaining sections of this paper to check, to what extent such numerical artifacts can affect the MD description of structural and dynamical properties of proteins.

Methods of quality assessment

In order to estimate the quality of the numerous approximations utilized by FAMUSAMM and to measure the gain of computational efficiency we have carried out a series of test simulations. We have applied two versions of our method, one employing the linear extrapolation, which we call FAMUSAMM/linear, and one using the DC-1d extrapolation scheme, which we correspondingly denote by FAMUSAMM/DC-1d. To enable comparisons with established methods, we have also performed test simulations using a cutoff method,[14] the fast SAMM algorithm[26] and the slow but exact evaluation of the Coulomb sum, eq. (1), which we use as the reference method.

As discussed in the introduction, the complicated and chaotic nature of protein dynamics renders quality assessments of approximate MD algorithms a non-trivial task. For example, comparisons of system trajectories or of other atomic details obtained from simulations carried out with different approximation schemes are useless for that purpose, since possible observed deviations merely reflect the chaotic character of these details and thus do not allow to derive accuracy measures. Instead, following the arguments and suggestions in refs. 16 and 20, we will apply a set of so-called "relevant" statistical observables for our intended estimates of algorithmic quality. These observables have been selected according to the conditions, that they should exhibit regular, i.e., non-chaotic temporal behavior and that they should refer to functionally important properties of proteins[16,20]

Model systems and test simulations

Our study is divided into two parts: The first part studies effects, which are observable within simulation times covering a few ps, and addresses questions like the accuracy of force approximation and the size of the energy drift caused by algorithmic noise. As our sample system we have chosen here the small protein *bovine pancreatic trypsin inhibitor* (BPTI), which is frequently used as a test system for evaluation of simulation methods.[53,7,54,3] We embedded BPTI in a water shell of 27 Å radius, comprising a total of 7,147 atoms.

The second part serves to estimate algorithmic accuracy with respect to effects on longer time scales. For that purpose the sample system introduced above is prohibitively large as far as the intended reference simulations are concerned, since these require the computationally very expensive evaluation of the Coulomb sum, eq. (1). Therefore we had to choose here a much smaller sample system, namely BPTI *in vacuo* comprising only 568 atoms. The test simulations from which we have extracted regular statistical observables for our quality estimates covered 1.2 ns each.

Since it is well known that for realistic protein simulations the natural environment has to be included[6,7,55,4] a note on our choice of an *in vacuo* test system is necessary: Concerning our long time simulations we do not aim to evaluate the quality of the physical model of the protein, which, here, may be poor. Rather, we want to check to what extent our algorithmic approximations affect the molecular dynamics of our given model. For that purpose, the physical model used here does not necessarily have to be very accurate as long as possible artifacts in realistic applications are likely to show up in our test simulations, too.

All test simulations were carried out with the MD program EGO_VIII,[51] which employs the CHARMM force field.[14] Non-polar hydrogen atoms were represented by compound atoms,[14] and the lengths of chemical bonds involving polar hydrogen atoms were fixed using the SHAKE algorithm.[3] An integration step size of 1 fs was used. Water molecules were described by the TIP3 model.[56] The interaction list was updated every 64-th integration step. For the cutoff simulations a cutoff distance of 9 Å and a switching function[57] were chosen. Translations and rotations of the protein were eliminated as described in refs. 58 and 59. Prior to the test simulations, BPTI *in vacuo* and the BPTI-water system were equilibrated for 50 ps and 70 ps, respectively, using the reference method. For the long-time simulations of BPTI *in vacuo* the temperature was kept fixed at 300 K by weakly coupling the system to a heat bath using a coupling constant of 10^{-13} s as defined in refs. 60 and 61. In contrast, the short-time simulations of the large BPTI-water system were carried out in the microcanonical ensemble as to enable an estimate, to what extent algorithmic artifacts entail violations of energy conservation.

For the first part of our quality estimates we will compare results of simulations obtained by the supposedly exact reference method with results obtained by SAMM, FAMUSAMM/linear and FAMUSAMM/DC-1d. Here, comparisons with the cutoff method are superfluous, since related accuracy tests comparing the various versions of SAMM with that method have been published previously.[19,18] By these tests the superiority of the SAMM algorithms as compared to the conventional cutoff procedure has been convincingly demonstrated. Thus, it solely remains to be checked, to what extent our FAMUSAMM approach preserves the advantages of its parent method.

For the second part we have performed 1.2 ns test simulations utilizing all methods under consideration. Here we have included the cutoff approach, since related investigations have not yet been published. Due to the statistical nature of our observables a second 1.2 ns test simulation was necessary for the reference method in order to enable an estimate of the sizes of statistical fluctuations. Only upon such estimates algorithmic artifacts can be identified and can be distinguished from inevitable fluctuations of the necessarily statistical observables compared. The initial conditions of the first reference simulation and of the test simulations for the approximate methods were chosen as the atomic positions and velocities obtained after equilibration. For the second reference simulation these initial positions were modified by extremely small random amounts. Note that the chaotic character of protein dynamics ensures complete decorrelation of the two reference simulations within a few picoseconds.

Short Time Observables

To measure the numerical accuracy of the force approximation we follow the suggestion in ref. 19. We consider the root mean square (*rms*) error of forces,

$$\eta(t) = \left[\frac{\sum_{i=1}^{N} \mid \mathbf{F}_i^{approx}(t) - \mathbf{F}_i^{ref}(t) \mid^2}{\sum_{i=1}^{N} \mid \mathbf{F}_i^{ref}(t) \mid^2} \right]^{1/2}, \tag{7}$$

where $\mathbf{F}_i^{approx}(t)$ and $\mathbf{F}_i^{ref}(t)$ denote the approximated and the exact Coulomb force, respectively, acting on atom i at time t, and where the forces are calculated using the trajectory of the reference simulation. As an accuracy measure we take the mean $\langle \eta \rangle$ of the *rms* error $\eta(t)$ for the simulated trajectory. The *rms* error of FAMUSAMM $\eta(t)$ will fluctuate in the course of a simulation, because, as is apparent from Figure 4, the accuracy of the force approximation should vary from step to step. The size of these fluctuations is measured by the standard deviation $\sigma_\eta = \langle (\eta - \langle \eta \rangle)^2 \rangle^{1/2}$.

Taking the point of view of numerical mathematics one might expect that $\langle \eta \rangle$ and σ_η represent measures for the algorithmic noise associated with a given approximation method. However, as discussed in refs. 16 and 20 there are physical considerations contradicting that expectation. For instance, the DC-1d extrapolation scheme has been designed to guarantee optimal energy conservation and other useful properties. Correspondingly, that method has been demonstrated to entail smaller algorithmic noise in the framework of pure force extrapolation than the linear extrapolation scheme, although it exhibits larger values of $\langle \eta \rangle$ and σ_η than the latter.[16, 20] It remains to be checked whether this is also the case in our framework of local Taylor expansion extrapolations. Therefore we also have to consider the size of uncontrolled algorithmic noise. In the microcanonical simulations at hand the estimation of that size is trivial. As algorithmic noise is the sole cause for energy transfer into such systems, one simply has to monitor the drift of the total energy $\Delta E_{total}/\Delta T$.

Long Time Observables

To estimate the influence of the applied approximations on the description of protein structure and dynamics, we have extracted the following observables from the long time simulations. As a measure for the temporal evolution of protein structure we have chosen the *rms* deviations $\rho_\alpha(t)$ of the intermediate configurations from the initial structure. If $\mathbf{r}_i(0)$ are the atomic positions obtained after the equilibration phase, the *rms* deviations of the set I_B of heavy backbone atoms ($\alpha = B$) and of the corresponding set I_S of side chain atoms ($\alpha = S$) are given by

$$\rho_\alpha(t) = \left[\frac{1}{N_\alpha} \sum_{i \in I_\alpha} \mid \mathbf{r}_i(t) - \mathbf{r}_i(0) \mid^2 \right]^{1/2} \quad , \tag{8}$$

where N_α denotes the number of atoms in the set I_α.

To estimate approximation effects on protein dynamics we monitor correlations between the motions of atoms i and j by the normalized covariance matrix

$$K_{ij} = \frac{\langle (\mathbf{r_i} - \langle \mathbf{r_i} \rangle) \cdot (\mathbf{r_j} - \langle \mathbf{r_j} \rangle) \rangle}{\left[\langle (\mathbf{r_i} - \langle \mathbf{r_i} \rangle)^2 \rangle \langle (\mathbf{r_j} - \langle \mathbf{r_j} \rangle)^2 \rangle \right]^{1/2}} \quad , \tag{9}$$

where the averages are taken over trajectory sections specified further below.

Subsequently we compare covariance matrices K_{ij}^A and K_{ij}^B from simulations A and B, respectively, both by a graphical method and by a single numerical observable. The latter is the *rms* difference of the cross correlations, i.e.,

$$\kappa = \left[\frac{\sum_{i<j} (K_{ij}^B - K_{ij}^A)^2}{\sum_{i<j} (K_{ij}^A)^2} \right]^{1/2} \quad , \tag{10}$$

whereas the former is an overlay of certain contour plots pertaining to the K_{ij}. These contour plots visualize histograms representing the frequencies of values $K_{ij} \in [-1, 1]$ as functions of atomic distances r_{ij} and will be called covariance plots.

As a technical point we would like to note that comparisons of the type sketched above require, that the trajectories refer to one and the same conformational substate of the protein considered. The reason is, that dynamical correlations may vary from substate to substate, such that differences of these correlations, which are caused by the application of different algorithms, may be obscured.

Results of quality assessment

In this section we will present the results of our study on the algorithmic accuracy of the FAMUSAMM method. First we will provide values for the short time observables extracted from the associated test simulations on the large BPTI-water system. Subsequently we will present values for the long time observables obtained for the simulations on BPTI *in vacuo.*

Short time observables

Table I compares the accuracy of force calculations achieved by the approximation schemes SAMM, FAMUSAMM/DC-1d and FAMUSAMM/linear, respectively. As accuracy measures the table displays the time averages $\langle \eta \rangle$ and standard deviations σ_η of the *rms* errors $\eta(t)$ given by eq. (7). These values have been extracted from trajectories of 0.4 ps duration. In addition, the values for the average energy drifts $\Delta E_{total}/\Delta T$ as calculated from trajectories of $\Delta T = 60$ ps duration, are given. Table I

Method	$\langle\eta\rangle$ [%]	σ_η [%]	$\Delta E_{total}/\Delta T$ [kcal/(mol ps)]
SAMM	0.70	0.01	4.8
FAMUSAMM/DC-1d	1.04	0.14	9.2
FAMUSAMM/linear	0.80	0.08	75

Table I: Approximation errors of force calculation for various methods as measured by the time average $\langle\eta\rangle$ of the *rms* error, eq. (7), and by its standard deviation σ_η; also given are average energy drifts $\Delta E_{total}/\Delta T$ reflecting algorithmic noise.

As can be seen, the average *rms* errors $\langle\eta\rangle$ of force approximation are of about the same size (1 %) for all three methods. The $\langle\eta\rangle$ values of the FAMUSAMM algorithms are only slightly larger than that of SAMM. Thus, the multiple-time-step extrapolation procedures do not seem to sizably reduce the quality of force approximation achieved by SAMM. In contrast, as shown in ref. 19, cutoff methods exhibit errors $\langle\eta\rangle$ which are larger by at least a factor of ten. Thus, as far as the quality of force approximation is concerned, the FAMUSAMM procedures essentially preserve the advantageous properties of SAMM.

But considering the fluctuations σ_η of the error $\eta(t)$, a distinct difference between SAMM and the FAMUSAMM schemes becomes apparent. Due to the application of the multiple-time-step extrapolation procedures, for FAMUSAMM the fluctuations σ_η are tenfold larger than for SAMM. Thus the question arises, whether the slightly larger values of $\langle\eta\rangle$ and the drastically increased values of σ_η for the FAMUSAMM methods are accompanied by comparable increases of algorithmic noise.

Consideration of the associated values for the energy drifts in Table I reveals a slight increase for FAMUSAMM/DC-1d as compared to SAMM which is in line with the corresponding increase of $\langle\eta\rangle$. In contrast, a dramatically enhanced algorithmic noise is apparent for the FAMUSAMM/linear approach although in that case $\langle\eta\rangle$ is nearly as small as for SAMM. Hence, the superiority of the DC-1d scheme as compared to the linear extrapolation is actually preserved despite the fact, that in FAMUSAMM extrapolations are applied to local Taylor expansion coefficients representing approximated forces instead to explicitly calculated exact forces. Finally note, that the σ_η values are uncorrelated to the observed algorithmic noise; this finding underlines the validity of the respective arguments presented in refs. 16 and 20.

Long time observables

As to investigate the influence of approximation schemes on the description of protein structure and dynamics we first analyze the temporal evolution of the *rms* deviations $\rho_\alpha(t)$ defined by eq. (8). The graphs shown in Figure 5 have been obtained from the various 1.2 ns simulations of BPTI *in vacuo*. The graphs represent smoothened versions of $\rho_\alpha(t)$ obtained by taking local averages of 5 ps width.

The *rms* deviations $\rho_S(t)$ of the side chain atoms are always larger than the *rms* deviations $\rho_B(t)$ of the backbone atoms, since side chains are much more flexible than

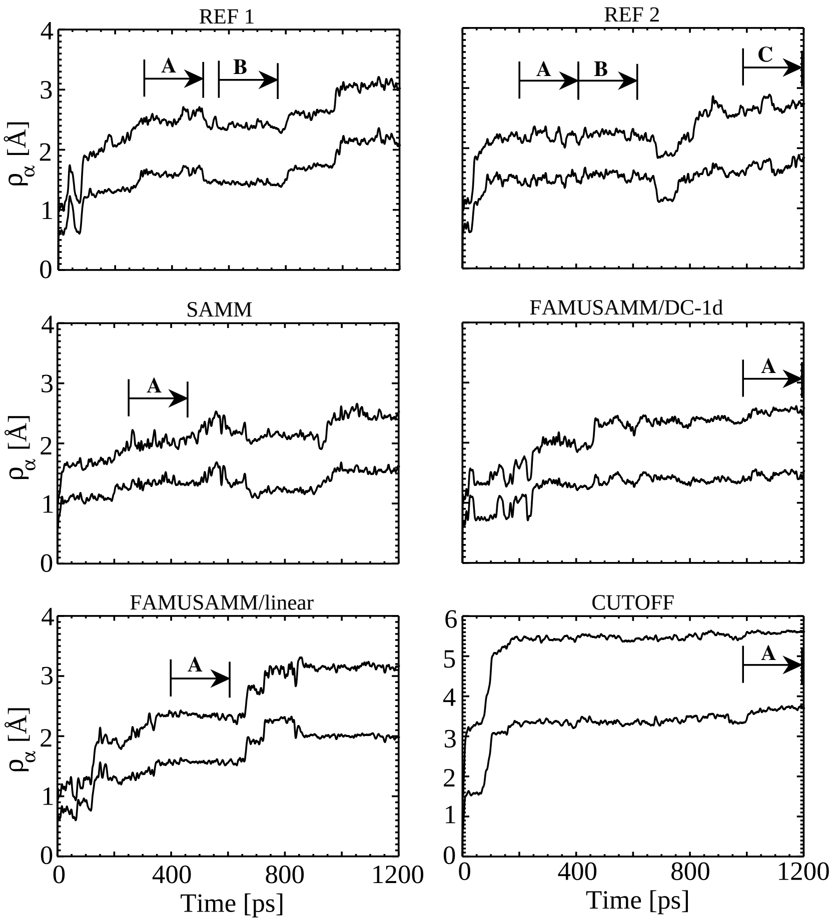

Figure 5: Temporal evolution of backbone *rms* deviations ρ_B and side chain *rms* deviations ρ_S for the two reference simulations (REF1 and REF2) and the four approximation methods; due to the flexibility of the side chains, ρ_S is always larger than ρ_B; trajectory sections, which correspond to stable substates and are used for later evaluation of the covariance matrices, are denoted by capital letters; for explanation see text.

the protein backbone. As can also be seen in Figure 5, both $\rho_B(t)$ and $\rho_S(t)$ increase with simulation time and generally exhibit a series of jumps. The first jump reflects the thermal fluctuations around the initial structure, whereas later jumps, as shown by a closer inspection of the associated structures (data not shown), arise from conformational transitions.

Furthermore one recognizes from the top two plots in Figure 5 that the time development of the *rms* deviations $\rho_\alpha(t)$ distinctly differs for the two reference simulations indicating that, indeed, the chaotic character of the system has led to a complete decorrelation of the respective trajectories. Yet the *rms* deviations resulting after 1.2 ns are of similar size and measure $\rho_B \approx 2\,\text{Å}$ and $\rho_S \approx 3\,\text{Å}$ respectively. Within the first 200 ps our *rms* deviations agree well with those obtained from other BPTI simulations *in vacuo* of up to 210 ps length.[6, 2]

Inspection of the subsequent three plots in Figure 5 shows that for SAMM and FAMUSAMM the overall behavior of $\rho_\alpha(t)$ is similar to that obtained for the two reference simulations. All graphs exhibit jumps pointing towards conformational transitions, and after 1.2 ns the values of $\rho_\alpha(t)$ are close to those of the reference simulations.

In contrast, $\rho_\alpha(t)$ exhibits a qualitatively and quantitatively different behavior for the cutoff simulation (lower right plot; note the different scale). Here one single and large conformational transition occurs after about 100 ps, and subsequently conformational transitions are suppressed[‖]. As shown by an analysis of the proteins structure, the transition was connected with a contraction upon which its overall size shrank by about 5 Å.

In summary, as monitored by $\rho_\alpha(t)$, no essential differences can be detected between the reference simulations, SAMM, and FAMUSAMM simulations, whereas the cutoff simulation apparently describes a different system.

We now turn to the investigation of dynamical correlations as described by the covariance matrices K_{ij} defined in eq. (9). As noted above, a comparison of such matrices requires that the trajectories, from which the K_{ij} are calculated, refer to identical or at least similar conformational substates of the protein. Inspection of Figure 5 shows that one can identify a couple of sections within the trajectories of the simulations pointing to conformational substates which are stable for at least 200 ps. These sections marked by capital letters in Figure 5 were employed for evaluation of associated covariance matrices.

Differences between the covariance matrices calculated in this way can be due to various sources: (i) they may represent pure statistical fluctuations, (ii) they may reflect dynamical variations associated with the individual conformational substates, and (iii) they may be caused by the application of deviating computational methods.

In order to disentangle these sources we first provide an estimate for (i) the size of statistical fluctuations. For that purpose we chose the two consecutive 200 ps sections of the second reference trajectory, which are marked in Figure 5, top right, as A and B. As is indicated by the constancy of the ρ_α in that period, these sections refer to one and the same conformational substate of BPTI. Figure 7(a) shows an overlay of the two covariance plots calculated from the covariance matrices associated with these trajectory sections and displays the value of the *rms* difference κ between the covariance matrices.

As one can see, the two plots are nearly identical. They show strong positive correlations between motions of neighboring atoms, which reflect the rigidity of the covalently bonded structure. At intermediate distances around 12 Å anti-correlations dominate, whereas correlations become positive again at distances larger than 25 Å. The *rms* deviation κ of the two cross correlations is 30 %. Thus deviations κ have to be distinctly larger than that value, and covariance plots have to exhibit much more pronounced differences than those in Figure 7(a), as to exclude mere effects of statistical fluctuations and to prove the existence of physical or algorithmic differences of the simulated dynamics.

A measure for the alteration of correlations of atomic motion, which is induced by a conformational transition into a different substate [see point (ii)], is provided by Figure 7(b). This figure also shows covariance matrices extracted from the second reference trajectory, but in contrast to Figure 7(a) trajectory sections referring to different substates

‖To assure this observation, we have performed a second cutoff simulation and have observed similar effects.

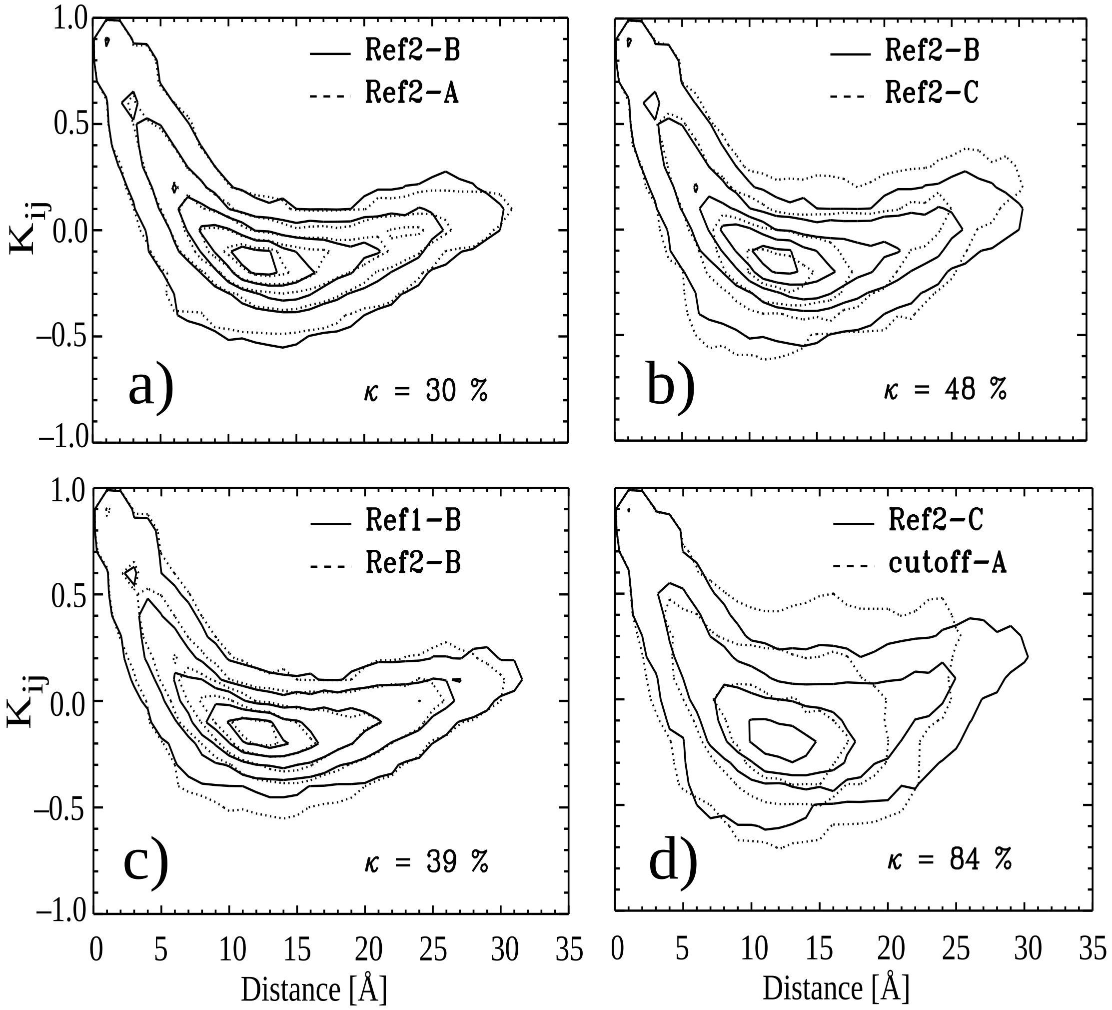

Figure 6: Covariance plots for comparison of BPTI dynamics within stable conformational substates simulated by different methods (see text for explanation).

of BPTI are compared. Specifically, sections B and C as marked in Figure 5, top right, have been chosen for comparison. In comparison to Figure 7(a), the covariance plots exhibit sizable differences, and the *rms* deviation of cross correlations, measuring $\kappa = 48\,\%$, is larger. But note that, apparently, the conformational transition from state B to state C has had little influence on the overall shape of the covariance plots.

Similar conclusions may be drawn, if one considers Figure 7(c). This figure compares covariance plots extracted from different trajectories calculated with the same computational method, i.e., sections B of the first reference simulation and section B of the second reference simulation as marked in Figure 5, top plots. These sections refer to slightly different substates, which have been selected according to a criterion of maximal structural similarity. Consequently, the good match of the covariance plots in Figure 7(c) as well as the low value of $\kappa = 39\,\%$ indicates that closely resembling structures exhibit similar dynamics.

In order to finally pin down the range within which the *rms* deviations κ of cross correlations can vary upon conformational transitions at a nanosecond time scale, we have additionally selected a whole set of 200 ps sections corresponding to stable substates from the two reference trajectories. We found that κ never exceeded 70 % and had a mean value of 58 %.

We now are in a position to use our long time observables as to distinguish algorithmic artifacts from effects of statistical fluctuations and conformational transitions. For this purpose we will select from the trajectories calculated with the approximate methods 200 ps sections with the following properties: (a) They should refer to stable substates, and (b) the structure of that substate should be as similar as possible to the structure of one of the substates sampled in one of the reference trajectories.

Figure 7(d) clearly indicates that the cutoff method is prone to algorithmic artifacts concerning the description of dynamics. This conclusion is validated in particular by the large value of 84 % for the *rms* deviation κ, which, in contrast to the covariance plot, solely expresses dynamical properties; note that the contour plots additionally display distance information.

Quite in contrast to the cutoff method all approximate methods, which account for long range Coulomb interactions by structure-adapted multipole expansions, show excellent performance in our test on dynamics. This statement is proven by Figure 8 displaying the respective covariance plots and the values of κ. As monitored by the plots and by κ, in all cases the substate dynamics calculated by SAMM and FAMUSAMM resembles quite closely to the dynamics of a similar substate sampled by the reference simulations. The comparisons are as close as those between the sections B of the reference simulations shown in Figure 7(c). Furthermore we have also selected a set of 200 ps sections referring to arbitrary substates from the SAMM and FAMUSAMM trajectories for comparison with substate sections from both reference simulations using κ as the observable. Like in the comparison among the reference sections noted further above, no value larger than 70 % was found for κ. As a consequence, concerning dynamics, no difference has been detected between the exact evaluation of long range Coulomb forces and our approximate methods.

Summarizing the results of our study on the accuracy of the various approximate MD algorithms we may state, that the FAMUSAMM/DC-1d algorithm (a) provides forces which are nearly as accurate as those obtained with SAMM, (b) entails only comparable little algorithmic noise, and (c) passes also our tests on the description of protein structure and dynamics with equally excellent results. In contrast, FAMUSAMM/linear is hampered by pronouncedly increased algorithmic noise, whereas the cutoff method exhibits unsatisfactory behavior in every respect. In view of these results we have discarded the linear multiple-time-step extrapolation scheme from further consideration within FAMUSAMM. It remains to be checked, to what extent the favorable algorithmic properties of FAMUSAMM are accompanied by a correspondingly favorable computation performance.

Computational Performance

Here we want to document that FAMUSAMM actually provides an enhanced computational efficiency both as compared to SAMM as well as to the reference method with

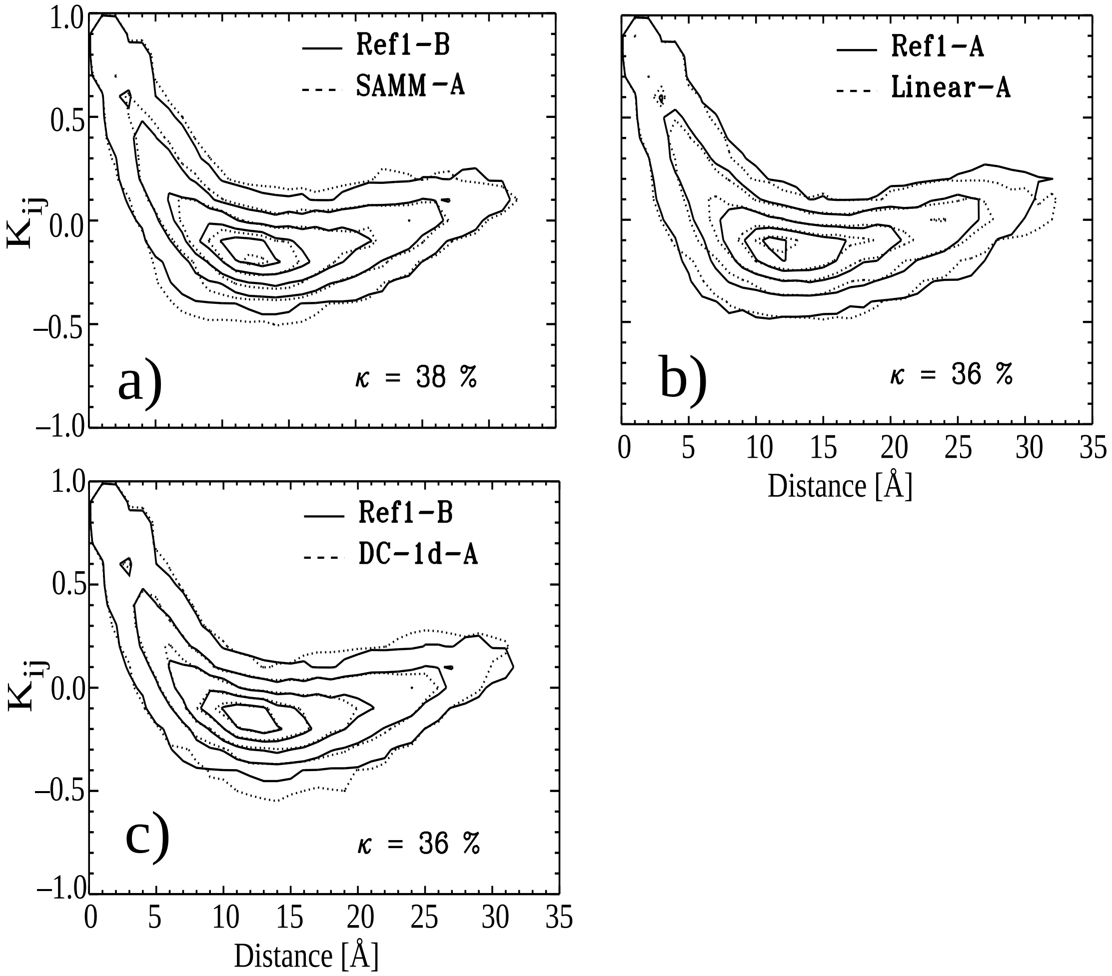

Figure 7: Further covariance plots for evaluation of our simulation methods.

its exact evaluation of the Coulomb sum. For that purpose we first consider how the computational effort of FAMUSAMM scales with system size.

For a quantitative performance assessment we have carried out a series of test simulations on systems of varying size using the sequential version of our MD program EGO_VIII. The chosen system sizes covered the range from 500 to 40,000 atoms. All simulations were executed on a DEC-ALPHA 3300L (175 MHz) workstation equipped with 96 MB RAM**. Figure 8 shows how the average computation time required for one MD integration step scales with system size. The choice of an average computation time for performance measurements is necessary, since in FAMUSAMM the time required for the force calculation varies from time step to time step (cf. Figure 4).

Figure 8 clearly shows that for systems comprising more than about 1,000 atoms FAMUSAMM achieves linear scaling of computational effort with size. The quadratic

**Due to limitations of memory we were unable to extend our tests to larger systems.

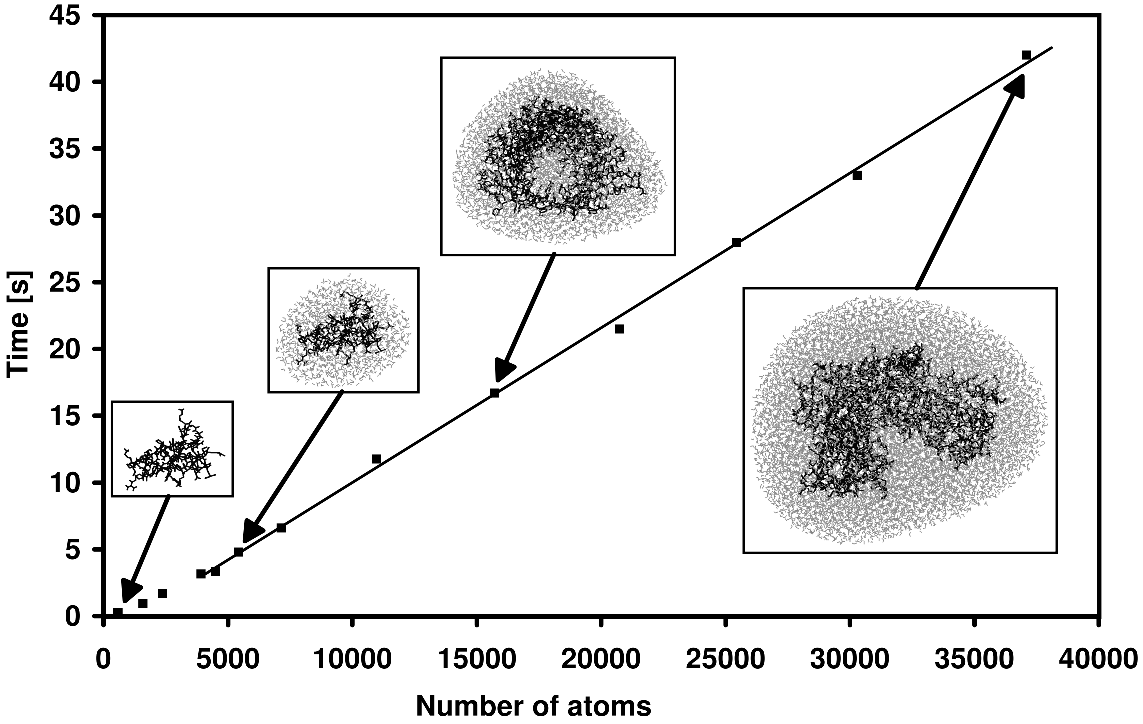

Figure 8: Average computation time for one step using EGO_VIII on a DEC-Alpha 3300L workstation (175 MHz) for simulation systems of varying size.

increase observed for very small systems is in line with the fact, that in the innermost distance class ($H = 0$ in Figure 4) forces are calculated by use of the Coulomb sum with its intrinsically quadratic scaling behavior. Thus, the theoretically expected linear scaling of FAMUSAMM for large systems is actually obtained in our implementation.

For performance comparisons with other methods consideration of system size is essential. As compared to its parent method SAMM, our new algorithm achieves a speed-up by a factor of *two* for small systems containing less than about $5,000$ atoms. This gain of efficiency is entirely due to our multiple-time-step extrapolation of forces in the innermost SAMM distance class $H = 0$ [see subsection c) in section "Combination of SAMM with a multiple-time-step method" as well as Figure 4]. Note that SAMM had been previously demonstrated to perform for systems up to that size as efficiently as the conventional cutoff methods.[26] Thus, FAMUSAMM beats the cutoff method in this regime also concerning efficiency.

For large systems comprising $36,000$ atoms FAMUSAMM performs *four* times faster than SAMM and as fast as cutoff. Here, the speed-up with respect to SAMM is essentially achieved by the multiple-time-step extrapolation of local Taylor expansions in the outer distance classes, and FAMUSAMM executes by a factor of 60 faster than the reference method based on the evaluation of the Coulomb sum.

Summary and Conclusion

In this paper we have combined the fast SAMM method for approximate evaluation of long range Coulomb interactions in MD simulations of protein dynamics with the DC-1d procedure designed for the same purpose into a new and highly efficient MD method. Upon extended test simulations the resulting FAMUSAMM algorithm has been demonstrated to preserve and combine the favorable properties of its parent methods concerning accuracy, lack of sizable algorithmic artifacts, and last, but not least, computational efficiency. Thus, FAMUSAMM opens the way for extended MD simulations which properly account for the long range Coulomb interactions of large protein-solvent systems comprising several $10,000$ atoms.

The algorithm has been implemented in the MD program EGO_VIII, which is based on the CHARMM force field, in a sequential and parallelized version and is available via Internet.[51] Whereas the current paper is exclusively devoted to the presentation and evaluation of our method, a previously published application[62] has already demonstrated its suitability for the quantitative analysis of experimental findings; there, the method has enabled an atomic interpretation of atomic force microscopy experiments on ligand-receptor binding.[63] As a check of the FAMUSAMM MD simulation performed in ref. 62 the *rms* deviation from the x-ray structure of a solvated protein (Streptavidin in water, 10969 atoms) has been monitored. After 0,5 ns simulation an *rms* deviation as small as $1,5$ Å was found showing that, indeed, FAMUSAMM provides realistic descriptions of solvated proteins.

Appendices

A: Pseudo-Code of FAMUSAMM

In this section we provide a summarizing pseudo-code description of FAMUSAMM. For clarity, we have omitted the special algorithmic steps, which are necessary during the start-up phase or are associated with hierarchical grouping and recalculation of the interaction list. As to decide, whether in a given time step forces or local Taylor expansions have to be explicitly calculated or may be extrapolated, we introduce a variable M, which denotes the highest hierarchy level H of SAMM for which an explicit calculation has to be performed (cf. Figure 4). M is called extrapolation level. The following abbreviations are used: LTE: local Taylor expansion, MM: multipole moments, SU: structural unit. Indentation defines the grouping of instructions.

For every integration step:
  Set extrapolation level M.
  Set all LTE coefficients to zero.
  Calculate MMs of all SUs (hierarchy level $H = 1$).
  For $H = 2$ to $H = H_{max}$:
    Calculate MMs for each cluster on higher hierarchy levels H.
  For $H = H_{max}$ down to $H = 1$:
    For every object at level H:
      If $H > M$:
        Extrapolate LTE of hierarchy level H.
      Else:
        Sum up contributions to LTE at level H according to interaction list
      If $H > 1$:
        Inherit LTE of level H to level $H - 1$.
  For every atom:
    Compute non-electrostatic forces forces (van der Waals, chem. binding forces).
    Add Coulomb forces from LTE of SU.
    Add Coulomb forces from atoms in distance class $j = 0$ (cf. Figure 4).
    If integration step even:
      Add calculated forces from distance class $j = 1$.
    Else:
      Add extrapolated forces from distance class $j = 1$.
    Perform Verlet integration step.

Acknowledgement:

This work has been supported by the Deutsche Forschungsgemeinschaft, grant SFB 143/C1.

References

[1] M. Levitt and S. Lifson, *J. Mol. Biol.*, **46**, 269 (1969).

[2] J. A. McCammon, B. R. Gelin, and M. Karplus, *Nature (London)*, **267**, 585 (1977).

[3] W. F. van Gunsteren and H. J. C. Berendsen, *Mol. Phys.*, **34**, 1311 (1977).

[4] O. Edholm, O. Berger, and F. Jähnig, *J. Mol. Biol.*, **250**, 94 (1995).

[5] H. Heller, M. Schaefer, and K. Schulten, *J. Phys. Chem.*, **97**, 8343 (1993).

[6] M. Levitt and R.Sharon, *Proc. Natl. Acad. Sci. USA*, **85**, 7557 (1988).

[7] M. Levitt, *Chem. Scr.*, **29A**, 197 (1989).

[8] M. C. Nuss, W. Zinth, W. Kaiser, E. Kölling, and D. Oesterhelt, *Chem. Phys. Lett.*, **117**, 1 (1985).

[9] G. Petsko, *Nature (London)*, **371**, 740 (1994).

[10] F. Zhou, A. Windemuth, and K. Schulten, *Biochem.*, **32**, 2291 (1993).

[11] M. P. Allen and D. Tildesley, *Computer Simulations of Liquids*, Clarendon, Oxford, 1987.

[12] J. A. McCammon and S. C. Harvey, *Dynamics of Proteins and Nucleic Acids*, Cambridge University Press, Cambridge, 1987.

[13] W. F. van Gunsteren and H. J. C. Berendsen, *Angew. Chem. Int. Ed. Engl.*, **29**, 992 (1990).

[14] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, *J. Comp. Chem.*, **4**, 187 (1983).

[15] C. L. Brooks III, B. M. Pettitt, and M. Karplus, *J. Chem. Phys.*, **83**, 5897 (1985).

[16] H. Grubmüller, *Molekulardynamik von Proteinen auf langen Zeitskalen*, Dissertation, Technische Universität München, Germany (1994).

[17] R. J. Loncharich and B. R. Brooks, *Proteins*, **6**, 32 (1989).

[18] C. Niedermeier, *Modellierung elektrostatischer Wechselwirkungen in Proteinen: Eine strukturadaptierte Multipolmethode*, Dissertation, Ludwig-Maximilians-Universität, München, Germany (1995).

[19] C. Niedermeier and P. Tavan, *J. Chem. Phys.*, **101**, 734 (1994).

[20] H. Grubmüller and P. Tavan, in preparation.

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

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

[23] L. Greengard and V. Rokhlin, *J. Comp. Phys.*, **73**, 325 (1987).

[24] L. Greengard and V. Rokhlin, *Chem. Scr.*, **29A**, 139 (1989).

[25] J. F. Leathrum and J. A. Board, *The parallel fast multipole algorithm in three dimensions*, Technical report, Dept. of Electrical Engineering, Duke University, Durham (1992).

[26] C. Niedermeier and P. Tavan, *Mol. Sim.*, **17**, 57 (1996).

[27] A. Ahmad and L. Cohen, *J. Comp. Phys.*, **12**, 389 (1973).

[28] W. B. Streett, D. J. Tildesley, and G. Saville, *Mol. Phys.*, **35**, 639 (1978).

[29] R. C. Y. Chin, G. W. Hedstrom, and F. A. Howes, *Considerations on Solving Problems with Multiple Scales*. Academic Press, Orlando, Florida, 1985.

[30] M. E. Tuckerman, G. J. Martyna, and B. J. Berne, *J. Chem. Phys.*, **93**, 1287 (1990).

[31] M. E. Tuckerman, B. J. Berne, and G. J. Martyna, *J. Chem. Phys.*, **94**, 6811 (1991).

[32] H. Grubmüller, H. Heller, A. Windemuth, and K. Schulten, *Mol. Sim.*, **6**, 121 (1991).

[33] R. D. Skeel and J. J. Biesiadecki, *Ann. Num. Math.*, **1**, 191 (1994).

[34] J. A. Board, Jr., J. W. Causey, J. F. Leathrum, Jr., A. Windemuth, and K. Schulten, *Chem. Phys. Lett.*, **198**, 89 (1992).

[35] H.-Q. Ding, N. Karasawa, and W. A. Goddard III, *J. Chem. Phys.*, **97**, 4309 (1992).

[36] M. O. Fenley, W. K. Olson, K. Chua, and A. H. Boschitsch, *J. Comp. Chem.*, **17**, 976 (1996).

[37] A. Windemuth, *Advanced Algorithms for Molecular Dynamics Simulation: The Program PMD.* ACS Books, 1995.

[38] R. Zhou and B. J. Berne, *J. Phys. Chem.*, **103**, 9444 (1995).

[39] P. P. Ewald, *Ann. Phys. (Leipzig)*, **IV**, 253 (1920).

[40] A. Y. Toukmaji and J. J. A. Board, *Computer Phys. Comm.*, **95**, 73 (1996).

[41] E. L. Pollock and J. Glosli, *Computer Phys. Comm.*, **95**, 93 (1996).

[42] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. G. Pedersen, *J. Chem. Phys.*, **103**, 8577 (1995).

[43] P. Procacci, T. Darden, and M. Marchi, *J. Phys. Chem.*, **100**, 10464 (1996).

[44] C. L. Brooks III and M. Karplus, *J. Chem. Phys.*, **79**, 6312 (1983).

[45] D. Beglov and B. Roux, *J. Chem. Phys.*, **100**, 9050 (1994).

[46] P. E. Smith and B. M. Pettitt, *J. Chem. Phys.*, **105**, 4289 (1996).

[47] H. Bekker, *Molecular Dynamics Simlation Methods Revised*, Dissertation, Rijksuniversiteit Groningen (1996).

[48] A. Brünger, C. Brooks, and M. Karplus, *Chem. Phys. Lett.*, **105**, 495 (1994).

[49] T. M. Martinetz and K. J. Schulten, In *Proceedings of the Int. Conf. on Artificial Neural Networks, ICANN-91, Espoo, Finland, 24–28 June 1991*, O. Simula, Ed., Elsevier Science Publishers, Amsterdam, 1991, p. 397.

[50] M. Eichinger, *Paralleler schneller Multipolalgorithmus mit Mehrschrittverfahren für Molekulardynamiksimulationen*, Diplomarbeit, Ludwig-Maximilians-Universität, München, Germany (1995).

[51] M. Eichinger, H. Grubmüller, and H. Heller, *User Manual for EGO_VIII, Release 2.0*, Theoretische Biophysik, Institut für Medizinische Optik, Ludwig-Maximilians-Universität, Theresienstr. 37, D-80333 München, Germany, 1995. Electronic access: http://www.imo.physik.uni-muenchen.de/ego.html.

[52] M. Eichinger, H. Grubmüller, and H. Heller (1997), in preparation.

[53] S. Hayward, A. Kitao, and N. Gō, *Physica Scripta*, **3**, 936 (1994).

[54] S. Swaminathan, T. Ichiye, W. van Gunsteren, and M. Karplus, *Biochem.*, **21**, 5230 (1982).

[55] M. Saito, *Mol. Sim.*, **8**, 321 (1992).

[56] W. L. Jorgensen, J. Chandrasekhar, and J. D. Madura, *J. Chem. Phys.*, **79**, 926 (1983).

[57] A. Brünger, *X-PLOR Manual*, The Howard Hughes Medical Institute and Department of Molecular Biophysics and Biochemistry, Yale University, New Haven, 1992.

[58] C. Eckart, *Phys. Rev.*, **47**, 552 (1935).

[59] N. Ehrenhofer, *Untersuchung der Konformationsdynamik eines vereinfachten Proteinmodells*, Diplomarbeit, Ludwig-Maximilians-Universität, München, Germany (1994).

[60] H. C. Anderson, *J. Chem. Phys.*, **72**, 2384 (1980).

[61] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak, *J. Chem. Phys.*, **81**, 3684 (1984).

[62] H. Grubmüller, B. Heymann, and P. Tavan, *Science*, **271**, 997 (1996).

[63] E.-L. Florin, V. T. Moy, and H. E. Gaub, *Science*, **264**, 415 (1994).