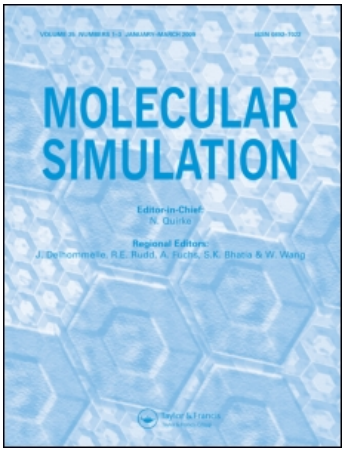

Molecular Simulation
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713644482

Rapid Calculation of Any Dielectric Function for Molecular Dynamics Simulations of Biological Macromolecules

M. Toulouse[a]; V. Fritsch[a]; E. Westhof[a]
[a] Institut de Biologie Moléculaire et Cellulaire du C.N.R.S., Strasbourg, Cedex, France

To cite this Article Toulouse, M. , Fritsch, V. and Westhof, E.(1992) 'Rapid Calculation of Any Dielectric Function for Molecular Dynamics Simulations of Biological Macromolecules', Molecular Simulation, 9: 3, 193 — 200
To link to this Article: DOI: 10.1080/08927029208047426
URL: http://dx.doi.org/10.1080/08927029208047426

Molecular Simulation, 1992, Vol. 9, pp. 193–200
Reprints available directly from the publisher
Photocopying permitted by license only

RAPID CALCULATION OF ANY DIELECTRIC FUNCTION FOR MOLECULAR DYNAMICS SIMULATIONS OF BIOLOGICAL MACROMOLECULES

M. TOULOUSE, V. FRITSCH and E. WESTHOF*

Institut de Biologie Moléculaire et Cellulaire du C.N.R.S., 15 rue R. Descartes, 67084 Strasbourg, Cedex, France

(*Received January 1992, accepted February 1992*)

Molecular dynamics simulations of macromolecules carrying charges, like nucleic acids, are hampered by proper treatment of the electrostatic problem. Although they can only constitute approximations, dielectric functions depending on the interatomic distance have the advantage of being easier to implement and faster to run than simulations with explicit solvent molecules and periodic boundary conditions. Further, their usefulness has been shown in several instances. Here, we describe a fast algorithm for calculating any dielectric function in MD simulations of biological macromolecules.

KEY WORDS: Electrostatics, dielectric functions, molecular dynamics, macromolecules

1 INTRODUCTION

The main theoretical bottleneck in molecular dynamics (MD) simulations of nucleic acids resides in the negative charge carried by the phosphate group of each nucleotide residue and the subsequent treatment of the electrostatics (for review, see [1]). Clearly, explicit water molecules with ions could be introduced in an all-encompassing simulation at the cost of a much longer set-up and computation time with possible coherence and convergence problems. Although ultimately such calculations have to be performed, the development of a fast and easily mastered computational approach for testing and screening nucleic acid models is still necessary and useful.

The potential energy function used in the modeling program AMBER 3.0 [2], including terms to describe the covalent structure deformations (bond stretches, bond angles deformations, torsional rotations) and terms to represent the nonbonded interactions broken in van der Waals, electrostatic and hydrogen bond contributions, has the following form

$$E = \sum_{\text{bonds}} k_d(d - d_0)^2 + \sum_{\text{angles}} k_\theta(\theta - \theta_0)^2 + \sum_{\text{dihedrals}} \frac{V_n}{2}[1 + \cos(n\phi - \gamma)]$$
$$+ \sum_{\text{nonbonded}} \left(\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} + 332\,\frac{q_i q_j}{\varepsilon r_{ij}}\right) + \sum_{\text{Hbonds}} \left(\frac{C_{ij}}{r_{ij}^{12}} - \frac{D_{ij}}{r_{ij}^{10}}\right).$$

In such an energy function, important errors can arise from the electrostatic term

*Corresponding author.

especially in the case of the negatively charged nucleic acids. For two partial charges q_1 and q_2 separated by an interatomic distance r (Å), the electrostatic term, in kcal · mol^{-1}, is given by:

$$V(r)_{\text{electrostatic}} = 332 \frac{q_1 q_2}{\varepsilon(r) r}$$

where $\varepsilon(r)$ is the dielectric function. The choice of a value $\varepsilon(r)$ is not easy and different approaches have been proposed (for review, see [1]). The program AMBER allows the use of $\varepsilon(r) = \alpha$ or $\varepsilon(r) = \alpha r$ with α scalar (usually, $\alpha = 1$ or $\alpha = 4$). Interested by the sigmoidal dielectric function proposed by Lavery *et al.* [3], we modified AMBER to allow the use of this distance dependent dielectric function, named ε_{cal}, in minimization studies and molecular dynamics simulations as well [4]. This sigmoidal function has the following form:

$$\varepsilon(r) = D - \frac{D - 1}{2} [(Ar)^2 + 2Ar + 2]e^{-Ar}$$

where $D = 78$, $A = 0.36$ [3] or $A = 0.16$ [5]. The first value gives the same dependence than the function developed by Hingerty *et al.* [6].

In the calculations we performed, the same form of the electrostatic potential was used for interactions between partial charges as for those interactions between the full charges on the anionic phosphate groups and ions. Although Friedman and Honig [7] have shown that electrostatic base-stacking interactions are adequately modelled using the Hingerty function, they concluded that phosphate–phosphate interactions could not be properly treated with such kind of dielectric model. The Debye–Hückel screening term should be also considered [8, 9]. However, the utility of the sigmoidal function [3] was shown in Monte Carlo calculations [10]. Similarly, after comparing 100 picoseconds (ps) MD simulations of small peptides, Daggett *et al.* [11] concluded that the sigmoidal function was much less computer intensive and more conformational space was sampled than with explicit water molecules. Solmajer and Mehler [12], using a slightly different distance dependent dielectric function [13], conclude also that the dynamic behaviour of the protein BPTI calculated with the screened Coulomb potential term compares more favorably with the time dependent structural changes calculated for the full system with explicit water molecules than with the time dependence obtained after a standard vacuum simulation. At the same time, the computer time was approximately an order of magnitude smaller.

After several conformational MD studies of poly(dA) · poly(dT), in which we compared the results obtained using $\varepsilon(r) = 4r$ and $\varepsilon(r) = \varepsilon_{\text{cal}}$, we found that the use of $\varepsilon(r) = \varepsilon_{\text{cal}}$ gave results in good agreement with experimental data [14, 15]. However, the simulation with the sigmoidal dielectric function took three times more computer time than $\varepsilon(r) = 4r$. Owing to the usefulness of the sigmoidal function, we decided to develop an algorithm which would allow the use of such a dielectric function without an increase in computation time.

2 DESCRIPTION OF THE ALGORITHM

A molecular dynamics simulation is performed by integrating Newton's equations of motion $\boldsymbol{F}_i = m_i \gamma_i$, where $\boldsymbol{F}_i$, the force on atom i of mass m_i is given by the negative

gradient of the total potential energy with respect to r_i:

$$F_i = -\frac{\partial}{\partial r_i} V(r_i)$$

For the electrostatic interactions, we have:

$$\begin{aligned} V(r)_{\text{electrostatic}} &= 332 \frac{q_1 q_2}{\varepsilon(r) r} \\ &= q_1 q_2 \mathscr{F}(r) \\ F(r)_{\text{electrostatic}} &= -\frac{\partial}{\partial r} V(r)_{\text{electrostatic}} \\ &= -\frac{q_1 q_2}{r\varepsilon(r)} \left(1 + \frac{r\varepsilon'(r)}{\varepsilon(r)}\right) \\ &= -q_1 q_2 \mathscr{G}(r) \\ \text{with} \quad \varepsilon'(r) &= \frac{\partial}{\partial r} \varepsilon(r) \end{aligned}$$

These calculations are done approximately N^2 times, for a system of N atoms.

After various trials, we settled for an algorithm which involves building a table for the functions $\mathscr{F}$ and $\mathscr{G}$. The array should contain $(1/r^2, \mathscr{F}(r), \mathscr{G}(r))$ for some real values r (in AMBER, the value transmitted is $1/r^2$). The first step is to find the values of r to store so that the errors on $\mathscr{F}$ and $\mathscr{G}$ are minimal. For some real value of $1/r^2$, the second step will consist in finding the element of the array which contains the nearest values of $\mathscr{F}$ and $\mathscr{G}$.

2.1 Construction of the Arrays

We divide the interval $[t, \mathscr{T}]$ in N parts $[r, R]$ such that:

- N is as small as possible.
- $|\mathscr{F}(R) - \mathscr{F}(r)| \leqslant e_0$, where e_0 is the greatest error we accept, equal to the precision of $\mathscr{F}(r)$, in percent (e_0 = (precision/100)$\mathscr{F}(r)$).

By the mean value theorem, since the function $\mathscr{F}$ is continuously smooth, there exists a $\xi \in [r, R]$ such that:

$$\mathscr{F}(R) - \mathscr{F}(r) = \mathscr{F}'(\xi)(R - r).$$

We search for R such that $|\mathscr{F}(R) - \mathscr{F}(r)| = e_0$, which is equivalent to $\mathscr{F}(R) - \mathscr{F}(r) = e_0$ since the function is increasing. R is such that $|\mathscr{F}'(\xi)(R - r)| = e_0$ which is equivalent to $\mathscr{F}'(\xi)(r - R) = e_0$ since $\mathscr{F}'$ is negative.

Assume that the interval $[r, R]$ is small enough so that the derivative $\mathscr{F}'$ of $\mathscr{F}$ is almost constant, we have $\mathscr{F}'(\xi) \simeq \mathscr{F}'(r)$. So, $e_0 = \mathscr{F}'(r)(R - r)$.

We then obtain:

$$\begin{aligned} R &= r - \frac{e_0}{\mathscr{F}'(r)} \\ &= r - \frac{\text{precision}}{100} \frac{\mathscr{F}(r)}{\mathscr{F}'(r)} \end{aligned}$$

$$= r + \frac{\text{precision}}{100} \frac{r\varepsilon(r)}{\varepsilon(r) + r\varepsilon'(r)}$$

The series $(r_n)_{n\in\mathbb{N}}$, defined by

$$\begin{cases} r_0 = t > 0 \\ r_{n+1} = r_n + \dfrac{\text{precision}}{100} \dfrac{r_n\varepsilon(r_n)}{\varepsilon(r_n) + r_n\varepsilon'(r_n)} \end{cases}$$

is built. It is easy to prove that this series is non negative, increasing and non bounded. Then for all non negative and non zero values of t and $\mathscr{T}$, the series will have a finite number of terms on the interval $[t, \mathscr{T}]$.

• *Algorithm*

$r := t$
$N := 1$
While $(r < \mathscr{T})$

Do
$$\begin{cases} Tabule[N, 1] = \dfrac{1}{r^2} \\ Tabule[N, 2] = \mathscr{F}(r) \\ Tabule[N, 3] = \mathscr{G}(r) \\ r = r * \left(1 + \dfrac{\text{precision}}{100} \dfrac{\varepsilon(r)}{\varepsilon(r) + r\varepsilon'(r)}\right) \\ N = N + 1 \end{cases}$$

$$Tabule[N, 1] = \frac{1}{\mathscr{T}^2}$$

$$Tabule[N, 2] = \mathscr{F}(\mathscr{T})$$

$$Tabule[N, 3] = \mathscr{G}(\mathscr{T})$$

With *precision* equal to 10, such a method divides the [1, 30] interval in 76 parts (interval in Å). In this case, we obtain an error less than 8% for $\mathscr{F}$, instead of the 10% allowed. We also note that with such a subdivision, only computed for $\mathscr{F}$, we obtain $\mathscr{G}$ an error of less than 12%. In case one is interested in smaller errors in $\mathscr{G}$ (for example use in a minimization algorithm), one should construct

$$R' = r - \frac{\text{precision}}{100} \frac{\mathscr{F}'(r)}{\mathscr{F}''(r)}$$

and choose the smallest value among R and R'.

2.2 Search for the Proper Value of the Running Index i

For the computations of functions $\mathscr{F}$ and $\mathscr{G}$ in AMBER, we get $R = 1/r^2$, and we have an array *Tabule* [ ] which contains the values of $\mathscr{F}$ and $\mathscr{G}$ for some reals $(R_i = 1/r_i^2,\ i = 1, \ldots, M)$.

We have then to find the index i such that $R_i \geqslant R > R_{i+1}$, and we will give to $\mathscr{F}$ and $\mathscr{G}$ the values *Tabule*[*i*, 2] and *Tabule*[*i*, 3].

Thus, given N values of R_i, we have to locate R among them. Since the classic shell

sorts are too slow, we look for a "simple function" which links a given value $R = 1/r^2$ to the index i such that $R_i \geqslant R > R_{i+1}$.

Given M values of R_i, we have to compare the *ndist* first decimals of the R_i to distinguish them; thus we calculate $N(R_i) = INT(10^{ndist} * R_i)$, which is a bijection between the set of the R_i values and a subset of integers.

Assume that $R \in]0, 1]$, we can calculate $N(R)$ and then two cases appear:

- 1st case: $R < R_M$, which corresponds to $r \geqslant r_M$, and then $0 \leqslant N(R) \leqslant N(R_M)$.
- 2nd case: $R \geqslant R_M$ then it exists $i \in \mathbb{N}$ such that $R_{i+1} < R \leqslant R_i$. Then we have $N(R_{i+1}) \leqslant N(R) \leqslant N(R_i)$.

Then we build the array *Nindex* [ ] of size $N(R_0)$, of which the p^{st} element holds the i value such that $N(R_{i+1}) < p \leqslant N(R_i)$. If such a value does not exist, *Nindex* $[p] = M$, following the examples

$$\begin{aligned} Nindex[N(R_i)] &= i \\ Nindex[N(R_1)] &= 1 \\ Nindex[N(R_M)] &= M \\ Nindex[N(1/\infty^2)] &= M \end{aligned}$$

Algorithm

$Nindex[1] := M$
For $i = 2$ **to** M, $Nindex\ [i] := 0$
For $i = 1$ **to** M
Do { $Num = 1 + INT(10^{ndist} * Tabule[i, 1])$; $Nindex[Num] = i$ }
For $i = 1$ **to** 10^{ndist}
Do { $Num = Nindex[i]$; **If** $(Num = 0)$ **Then** $Nindex[i] = Nindex[i - 1]$ }

The index is moved one unit in order to replace $N(\infty)$, which is zero, by 1.

2.3 Implementation

Algorithm

- Build of the arrays *Tabule*[ ] and *Nindex*[ ] at the begining of the program
- Obtain $\mathscr{F}$ et $\mathscr{G}$ by doing:

$$\begin{aligned} Num &= 1 + INT(10^{ndist} * 1/r^2) \\ Num &= Nindex[Num] \\ \mathscr{F} &= Tabule[Num, 2] \\ \mathscr{G} &= Tabule[Num, 3] \end{aligned}$$

2.4 Increase in Precision

We have:

$$R = r + \frac{\text{precision}}{100} \frac{r\varepsilon(r)}{\varepsilon(r) + r\varepsilon'(r)}$$

$$-Num = 1 + INT(10^{ndist} * R)$$

To increase the precision, it is enough to decrease the term *precision* which is the maximal error (in percent) allowed. But, the terms *ndist* and *precision* are strongly correlated; the *ndist* has to distinguish the boundaries, so it has to be all the more "spreading" than they are to each other.

For instance, for the interval [1, 30] and for a precision equal to 5%, 10^{ndist} equal to 10 000 is sufficient, but for a precision equal to 1%, 10^{ndist} must be ten times greater (*ndist* should be adapted to the *precision* given by the user and, consequently for safety, the size of *Nindex*[ ] should be overevaluated). The size of *Tabule*[ ] should be also considered since it increases with the precision as the number of intervals increases.

For the interval [1, 30] and for a precision (in percent) of:

precision	size of Tabule [ ]	10^{ndist}
10	76	10 000
5	156	10 000
1	770	100 000

Note that the size of Tabule[ ], as function of the precision, has almost a linear variation (see Table 1).

3. CONCLUSION

Applied to MD simulations of poly(dA) · poly(dT) (638 atoms), for 50 ps, with a precision of 1%, this algorithm reduces the computation time by 47% on a IBM 3090 (it took 696 min of CPU using the sigmoidal function, without this algorithm, and only 324 minutes with this algorithm). A 50% decrease in CPU time was also obtained for 50 ps MD simulations of the 17 bases of the anticodon hairpin of yeast tRNA-asp (546 atoms). In that case, the MD simulations with explicit water molecules (1706 water molecules) run about 20 times longer [16].

A major advantage of the proposed algorithm is that it can easily be applied to any function with no increase in CPU time. Indeed, as discussed in the introduction, several levels of approximations can be introduced to try to improve the representation of electrostatic interactions. A constant, f_M, can be added to take care of ion condensation [17]. Atoms charges are often reduced by f_M to represent counterion screening effect in nucleic acid simulations. A second term, $(e^{\kappa\sigma}/1 + \kappa\sigma)e^{-\kappa r}$, represents the Debye–Hückel screening effect duc to the solvent polarizability. In this case, the electrostatic energy is reduced by interactions between atom charges and solvent dipoles. The electrostatic potential then becomes:

$$V(r)_{\text{electrostatic}} = 332 \frac{q_1 q_2}{\varepsilon(r) r} f_M \frac{e^{\kappa\sigma}}{1 + \kappa\sigma} e^{-\kappa r}$$

Table 1 Values of $\mathcal{F}(r)$ and $\mathcal{G}(r)$ obtained with a precision equal to 1% on $\mathcal{F}(r)$ in the interval [1, 30]. $\mathcal{F}_{cal}$ (respectively $\mathcal{G}_{cal}$) represents the explicit calculation of $\mathcal{F}(r)$ (respectively $\mathcal{G}(r)$), and $\mathcal{F}_{approx}$ (respectively $\mathcal{G}_{approx}$) gives the value obtained using the algorithm developed here. The difference between the two terms is represented, in percent, by $err_{\mathcal{F}}$ (respectively $err_{\mathcal{G}}$). Despite the fact that no condition on the precision of $\mathcal{G}(r)$ is imposed, $err_{\mathcal{G}}$ is very close to $err_{\mathcal{F}}$

r	$\frac{1}{r^2}$	$\mathcal{F}_{cal}$	$\mathcal{F}_{approx}$	$err_{\mathcal{F}}$	$\mathcal{G}_{cal}$	$\mathcal{G}_{approx}$	$err_{\mathcal{G}}$
1.000	1.000000	0.6914502382	0.6914502382	0.00	1.2754902840	1.2754902840	0.00
2.450	0.166597	0.0740828365	0.0737691000	0.42	0.2181289792	0.2172425836	0.41
3.900	0.065746	0.0187930558	0.0187905226	0.01	0.0542855598	0.0542777628	0.01
5.350	0.034938	0.0077983956	0.0077320538	0.85	0.0207043551	0.0205050092	0.96
6.800	0.021626	0.0042474535	0.0042083878	0.92	0.0102059813	0.0100932885	1.10
8.250	0.014692	0.0027303942	0.0027143650	0.59	0.0059116161	0.0058673806	0.75
9.700	0.010628	0.0019563448	0.0019539788	0.12	0.0038175168	0.0038112737	0.16
11.150	0.008044	0.0015102184	0.0015084241	0.12	0.0026649176	0.0026603995	0.17
12.600	0.006299	0.0012291735	0.0012240496	0.42	0.0019730141	0.0019607451	0.62
14.050	0.005066	0.0010396203	0.0010337072	0.57	0.0015301976	0.0015168053	0.88
15.500	0.004162	0.0009046136	0.0008994494	0.57	0.0012326991	0.0012217074	0.89
16.950	0.003481	0.0008040502	0.0007983691	0.71	0.0010250849	0.0010138038	1.10
18.400	0.002954	0.0007262796	0.0007228801	0.47	0.0008756194	0.0008693494	0.72
19.850	0.002538	0.0006642016	0.0006610631	0.47	0.0007650851	0.0007597316	0.70
21.300	0.002204	0.0006133096	0.0006105619	0.45	0.0006813140	0.0006769813	0.64
22.750	0.001932	0.0005706422	0.0005695415	0.19	0.0006163294	0.0006147185	0.26
24.200	0.001708	0.0005341957	0.0005312690	0.55	0.0005647650	0.0005607817	0.71
25.650	0.001520	0.0005025769	0.0005005109	0.41	0.0005229406	0.0005202988	0.51
27.100	0.001362	0.0004747924	0.0004715275	0.69	0.0004882966	0.0004843470	0.81
28.550	0.001227	0.0004501170	0.0004486569	0.32	0.0004590325	0.0004573413	0.37
30.000	0.001111	0.0004280094	0.0004280094	0.00	0.0004338700	0.0004338700	0.00

where the dielectric function $\varepsilon(r)$ accounts for local water structure and saturation of the water dipoles. Using the algorithm described here, the inclusion of all those terms will not increase the computer time.

References

[1] S.C. Harvey, "Treatment of electrostatic effects in macromolecular modeling", *Proteins*, **5**, 78 (1989).

[2] S. J. Weiner, P.A. Kollman, D.T. Nguyen and D.A. Case, "An all atom force field for simulations of proteins and nucleic acids", *J. Comput. Chem.*, **7**, 230 (1986).

[3] R. Lavery, H. Sklenar, K. Zakrzewska and B. Pullman, "The flexibility of the nucleic acids: (II) the calculation of internal energy and applications to mononucleotide repeat DNA", *J. Biomol. Struct. Dyn.*, **3**, 989 (1986).

[4] V. Fritsch and E. Westhof, "Minimization and molecular dynamics studies of guanosine and Z-DNA modified by N-2-acetylaminofluorene", *J. Comput. Chem.*, **12**, 147 (1991).

[5] J. Ramstein and R. Lavery, "Energetic coupling between DNA bending and base pair opening", *Proc. Natl. Acad. Sci. USA*, **85**, 7231 (1988).

[6] B.E. Hingerty, R.H. Ritchie, T.L. Ferrell and J.E. Turner, "Dielectric effects in biopolymers: the theory of ionic saturation revisited", *Biopolymers*, **24**, 427 (1985).

[7] R.A. Friedman and B. Honig, "The electrostatic contribution to DNA base-stacking interactions", *Biopolymers*, in press.

[8] M.O. Fenley, G.S. Manning and W.K. Olson, "Approach to the limit of counterion condensation", *Biopolymers*, **30**, 1191 (1990).

[9] M.O. Fenley, G.S. Manning and W.K. Olson, "A numerical counterion condensation analysis of the B–Z transition of DNA", *Biopolymers*, **30**, 1205 (1990).

[10] B. Jayaram, S. Swaminathan, D.L. Beveridge, K. Sharp and B. Honig, "Monte Carlo simulation studies on the structure of the counterion atmosphere of B-DNA. Variations on the primitive dielectric model", *Macromolecules*, **23**, 3156 (1990).

[11] V. Daggett, P.A. Kollman and I.D. Kuntz, "Molecular Dynamics simulations of small peptides: dependence on dielectric model and pH", *Biopolymers*, **31,** 285 (1991).
[12] T. Solmajer and E.L. Mehler, "Electrostatic screening in molecular dynamics simulations", in *Advances Biomolecular Simulations*, R. Lavery, J.-L. Rivail and J. Smith, (eds) (American Institute of Physics, New York, 1991), p. 115.
[13] E.L. Mehler and G. Eichele, "Electrostatic effects in water-accessible regions of proteins", *Biochemistry*, **23,** 3887 (1984).
[14] V. Fritsch and E. Westhof, "Three-center hydrogen bonds in DNA: molecular dynamics of poly-(dA) · poly(dT)", *J. Am. Chem. Soc.*, **113,** 8271 (1991).
[15] S. Brahms, V. Fritsch, J.G. Brahms and E. Westhof, "Investigations on the dynamic structure of adenosine and thymine containing DNA", *J. Mol. Biol*, in press.
[16] C. Rubin-Carrez and E. Westhof, in preparation.
[17] G.S. Manning, "The molecular theory of polyelectrolyte solutions with applications to the electrostatic properties of polynucleotides", *Quat. Rev. Biophys.*, **II,** 179 (1978).