

This article was downloaded by:

On: 14 January 2011

Access details: *Access Details: Free Access*

Publisher *Taylor & Francis*

Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



Molecular Simulation

Publication details, including instructions for authors and subscription information:

<http://www.informaworld.com/smpp/title~content=t713644482>

Parameterization of Coulomb interaction in three-dimensional periodic systems

G. Tóth^a; A. Vrabec^a

^a Institute of Chemistry, Loránd Eötvös University, Budapest, Hungary

To cite this Article Tóth, G. and Vrabec, A.(2007) 'Parameterization of Coulomb interaction in three-dimensional periodic systems', *Molecular Simulation*, 33: 13, 1033 — 1044

To link to this Article: DOI: 10.1080/08927020701579345

URL: <http://dx.doi.org/10.1080/08927020701579345>

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: <http://www.informaworld.com/terms-and-conditions-of-access.pdf>

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

Parameterization of Coulomb interaction in three-dimensional periodic systems

G. TÓTH* and A. VRABECZ

Institute of Chemistry, Loránd Eötvös University, P.O. Box 32, Budapest H-1518, Hungary

(Received June 2007; in final form July 2007)

A simple method is proposed to calculate Coulomb interactions in three-dimensional periodic cubic systems. It is based on the parameterization of the interaction on polynomials and rational functions. The parameterized functions are compared to tabulation methods, to the Ewald calculations and cubic harmonic function fits found in the literature. Our parameterizations are computationally more efficient than, the use of tabulations at all cases and seem to be more efficient than the cubic harmonic parameterizations in the case of simultaneous potential energy and force calculations. In comparison to the Ewald method, it is feasible to use the parameterizations on small systems and on systems, where pair-wise additive short-range interactions are dominant. One also may prefer the parameterizations to the Ewald method for large systems, if limited accuracy is needed. The embedding of the method into existing molecular dynamics and Monte Carlo simulation codes is very simple. The presented investigation contains some numerical experimental data to support the correct theoretical partition of potential energy in periodic systems, as well.

Keywords: Periodic systems; Long-range interactions; Coulomb interaction; Molecular dynamics; Monte Carlo simulation; Cubic harmonics

1. Introduction

The calculation of long-range interactions, e.g. Coulomb interactions, in periodic systems is a long-standing question in different areas of physics. The progress in computational modeling reassigned it partly to a numerical mathematical task, but the different physical ideas still play important roles.

For the energy calculation of a long-range potential, the trivial method is to, sum the interactions among all particles in all cells of the systems. If N denotes the number of the cells taken into account, and n is the number of the particles within one cell, the trivial energy calculation of the total system scales with N^2n^2 . If one uses the symmetry of the periodic images, and one is interested in the potential energy of a central cell, as it is schematically shown in figure 1, the trivial calculation scales with Nn^2 . This calculation is unfeasible, because one should apply a large number of image cells. A recommended minimal choice of N is a few thousand. The first computationally essential milestone in making it feasible was the method of Ewald [1]. He suggested, recomposing the interactions into two main parts.

The direct sum is given by the sum of screened interactions, using Gaussian-screening functions centered at each point charge. The reciprocal sum is given by the Fourier series representation of the solution to Poisson's equation in periodic boundary conditions, using the sum of the compensating Gaussians, again centered at each point charge, as a source distribution. The terms can be expressed analytically in these Fourier series [1,2].

The spreading of computer simulations demanded many new approaches, especially on the field of liquid matter science. Without going into details and classifying the methods, the most important ones are the reaction-field method [3,4], the different mesh methods [5–7] and the multi-pole expansions [8–10]. The description of these techniques can be found in the original publications, later applications and in textbooks [11–13] as well. There are numerous comparative studies, whereof we give references only to some traditional articles [14–16]. The methods scale with n differently, but in practical applications not the scaling, but the overall computational cost and accuracy are the most important factors at a given system size. For example, there is an n scaling method

*Corresponding author. Email: toth@chem.elte.hu

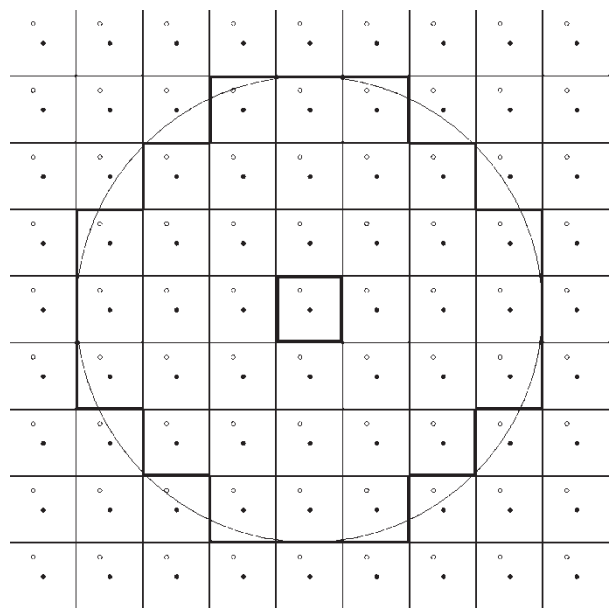


Figure 1. Schematic representation of a periodic system. The central cell and a possible choice of an N cell subsystem are denoted by thick borders.

[10], but the overall computational cost becomes beneficial only for large systems.

One can find also methods derived only using practical and computational assumptions instead of physical background. The simplest is the decomposition of the interaction into pair-interactions and tabulating the potential function in a three-dimensional grid. It is possible also to parameterize the tabulated grid using appropriate functions. This technique was applied in the historical work of Brush *et al.* [17] on one-component plasma. Hansen [18] used cubic harmonic functions (see also kubic harmonics) in the tabulation of interactions. The coefficients of this type of series expansion and the efficient choice of the function series were revisited and discussed several times [19–21]. The main advantage of the cubic harmonic functions is the natural possibility of describing cubic symmetry. The coefficients and the corresponding functions of Adams and Dubey [20] seem to be a practically feasible way in the calculation of Coulomb interactions.

The choice of the method depends on many factors. Practically, the system size is not the most important factor. In the case of classical mechanical molecule models, it is not necessary to assign charges to all species. In this case, the computational cost of the short-range interactions can become dominant against that one of the long-range interactions. The organization of the short-range interaction calculation differs from the long-range one. They are calculated on interaction pairs via one by one, while the Coulomb interactions are calculated for the whole system simultaneously in the Ewald method. Therefore, if the number of particles involved only in short-range interactions is significant, the program

structure should be optimized for the short-range interactions. Another factor is the simplicity of the code. For a program code developed for a few applications in a specific research project, it is not necessary to optimize ultimately for the speed performance, it is more important to write the code quickly and to be able to eliminate programming failures.

The purpose of our study, was to develop a very simple method for the calculation of Coulomb interactions in three-dimensional periodic systems. Simple means that it could be embedded easily into codes written for short-range interactions and it could be at least as simple and efficient than, the use of cubic harmonics. Of course, we had to suggest a method that is feasible in the side of calculation cost, as well.

2. Partitioning of the potential energy in periodic systems

The primary aim of our investigation was to parameterize the periodic Coulomb interactions. Before we started the calculations, we checked how the energy of one cell in a periodic system should be calculated. This is not only a case of an arbitrary definition, because the calculated potential energies are used very often. One example, is classical mechanical Monte Carlo simulation, where the energies appear in the probability factor. An incorrect energy definition doubles or halves the apparent temperature of the system. The question of the energy partition is also important in micro-canonical molecular dynamic simulations on periodic systems. We discuss this question, in this section. We know that, its content is slightly dissimilar to the main aim of our article.

Let us consider a periodic system. The particles interact with a pair-wise additive potential, where the potential depends solely on the distance of the particles. The simulation cell contains n particles. We define a system of all together $N - 1$ periodic images of the central cell and the central one. The schematic visualization of an N cell system can be seen in figure 1. The image cells are included in the system, if the center of an image cell is within a given distance from the central cell (figure 1). The potential energy of the total N cell system (U^{tot}), e.g. the electrostatic energy defined by the Coulomb interaction, can be decomposed into three terms: the interactions within the cells (intra term), the interactions among the images of the same particles in the different cells (self term) and the interactions of the distinct particles in the different cells (distinct term).

$$U^{\text{tot}} = \frac{1}{2} \sum_{k=1}^N \sum_{i=1}^n \sum_{j \neq i}^n u_{ij}^{kk} + \frac{1}{2} \sum_{k=1}^N \sum_{l \neq k}^N \sum_{i=1}^n u_{ii}^{kl} + \frac{1}{2} \sum_{k=1}^N \sum_{l \neq k}^N \sum_{i=1}^n \sum_{j \neq i}^n u_{ij}^{kl} \quad (1)$$

where u_{ij}^{kl} denotes the interaction between the i -th particle in the k -th cell and the j -th particle in the l -th cell. One can count the intra-, self- and distinct interactions:

$$\frac{1}{2}Nn(n-1) \quad \frac{1}{2}N(N-1)n \quad \frac{1}{2}N(N-1)n(n-1)$$

If one collects the interactions where particle one and all of its images are involved, one gets

$$U_1^{\text{all}} = \sum_{k=1}^N \sum_{j \neq 1}^n u_{1j}^{kk} + \frac{1}{2} \sum_k \sum_{l \neq k}^N \sum_{j \neq 1}^n u_{1j}^{kl} + \sum_k \sum_{l \neq k}^N \sum_{j \neq 1}^n u_{1j}^{kl} \quad (2)$$

and

$$N(n-1) \quad \frac{1}{2}N(N-1) \quad N(N-1)(n-1)$$

It is assumed that, $u_{ij}^{kl} = u_{ji}^{lk}$ in the last term. If the collection is restricted to the interaction of the first particle in the first cell:

$$U_1^1 = \sum_{j \neq 1}^n u_{1j}^{11} + \sum_{l \neq 1}^N \sum_{i \neq 1}^n u_{1i}^{l1} + \sum_{l \neq 1}^N \sum_{i \neq 1}^n u_{1i}^{l1} \quad (3)$$

and

$$(n-1) \quad (N-1) \quad (N-1)(n-1)$$

The last equation corresponds to all interactions involved the first cell explicitly:

$$U^1 = \frac{1}{2} \sum_{i=1}^n \sum_{j \neq i}^n u_{ij}^{11} + \sum_{l \neq 1}^N \sum_{i=1}^n \sum_{j \neq i}^n u_{ij}^{l1} + \sum_{l \neq 1}^N \sum_{i=1}^n \sum_{j \neq i}^n u_{ij}^{l1} \quad (4)$$

and

$$\frac{1}{2}n(n-1) \quad (N-1)n \quad (N-1)n(n-1)$$

The energy of one cell, e.g. in the Ewald sum, is defined as U^{tot}/N in most of the textbooks [11–13]. On contrary, in most of the lectures at conferences, if one is interested solely on the central cell, the energy of equation (4) is frequently used. Another introduction of the potential energy of periodic systems uses the formulae of electric fields. The effect of charges in the central cell and in the surrounding ones is expressed in an electrostatic potential. The equation is simply

$$U^1 = \frac{1}{2} \sum_{i=1}^n q_i \varphi(\mathbf{r}_i) \quad (5)$$

where q_i is the charge of the particle and $\varphi(\mathbf{r}_i)$ is the electrostatic potential at the \mathbf{r}_i point. Equation (5) differs from the usual definition of the energy of point charges by the factor of 1/2. Sometimes the factor is missing in equation (5). In this case, the 1/2 shifted into the calculation equation of the electrostatic potential.

If one checks the literature, there is a lack of detailed explanation on the role of the 1/2 factors. One may have an intuition of it coming from the pair-wise additive potential and/or of the constraint connecting the central and the

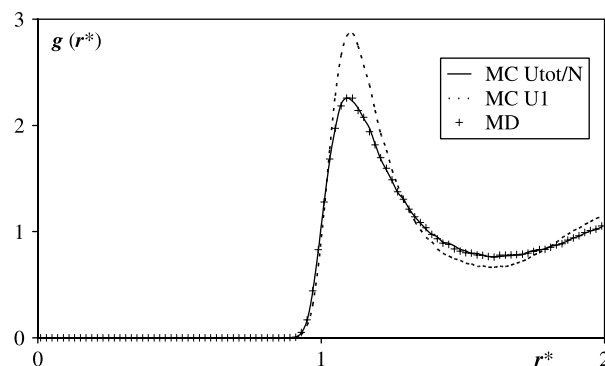


Figure 2. Pair-correlation functions of Lennard-Jones systems to check the partition choice of potential energy. The system consisted of 40 Lennard-Jones particles close to the triple point. r^* is in reduced unit [11].

image particles. If one compares equations (1) and (4), it can be concluded that the 1/2 factor appears in the intra term, if one is interested in all particles within at least the cell. The 1/2 multiplier of the self term is necessary, if one takes into account the periodic images. The 1/2 factor of the distinct term appears only if all particles in all cells are concerned.

At first, we checked the possible theoretical reasons and proofs, but none of them seemed to be helpful with complete certainty. Neither theoretician colleagues were able to provide simple convincing arguments. Therefore, we performed simple numerical experiments to choose the correct partition. It is known, that Monte Carlo and molecular dynamic simulations provide the same results in accordance the equivalence of the time and ensemble averages. The force acting on a particle and on all of its images is straightforwardly given by the Newtonian laws, it is the appropriate derivate of equation (2) divided by N . We performed molecular dynamics simulation where $n = 20 - 100$ and the particles interact with Lennard-Jones potential. This potential is not a long-range interaction, but in the case of small number of particles, its effective range exceeds the size of the cell. Simultaneously, we performed Monte Carlo simulation on the same systems, where we used alternatively U^{tot}/N or U^1 to calculate the potential energy. Our calculations showed numerically, that the U^{tot}/N definition is the correct one. The Monte Carlo and the molecular dynamics simulations provided the same structural results, only in this case. The pair-correlation functions of a molecular dynamics, a Monte Carlo simulation with U^{tot}/N , and a Monte Carlo one with U^1 are shown in figure 2. The energy conservation was satisfied also for the U^{tot}/N definition in the micro-canonical molecular dynamics simulation.

3. Details of the calculations

Let us recur to the main aim of our study: to the parameterization of the Coulomb interaction. We describe

the data sets, the formulae and the methods applied for the parameterization in this section. We put two particles with opposite unit charges in the central cell of a system like in figure 1. We centered the whole system on the position of the first particle. It meant, we had one particle in the origin and another somewhere in the cell. We defined a Coulomb-like interaction $u_{ij}^{kl} = z_i z_j / r_{ij}^{kl}$, where r_{ij}^{kl} is the scalar distance of the i -th particle in the k -th cell and the j -th particle in the l -th one. For the sake of simplicity, we defined all variables to be dimensionless, e.g. they were normalized by the edge length of the cell and by the magnitude of the charges. In the cases of parameterization and interpolation methods, the reduction of the space was feasible, wherein the process would be performed. One could use many simplifications for isotropic systems. If one puts the origin at the center of the central cell, one could define eight sub-cells, where only the sign of the second particle coordinates differed. The overall energy of the system was the same for all the eight positions, there were differences only in the sign of the forces. Furthermore, the choice of the axis was occasional. One might rotate the axis to define the coordinates of the second particle always to be $z \leq y \leq x$. It reduced the important space inside the cubic to its $1/6$. If we used both the sign invariance and the choice of axis one, it was enough to fit or interpolate the interaction in a $1/48$ part of the original cell. The shape of this reduced space was a tetrahedron inside the cubic central cell. Of course, if we used this reduced space in a real calculation, e.g. during a molecular simulation, we had to keep on file the original sign and order of the coordinates and we had to calculate the forces due to the original signs and axis.

The reference data were, calculated both by the trivial method and by the Ewald one as it is implemented in Ref. [11]. The difference was rather small. It was below $10^{-4}\%$, if we chose the maximum of the inverse-space sums large enough and we used an appropriate width for the Gaussian-distributions in the Ewald method, and if we calculated the trivial sum over a large number of image cells. In the presented calculations, we used 25 for the maximum of the square of the direct space sums and $L/5$ for the Gaussian width (see e.g. Refs. [11,13]) where L denotes the edge of the cell. ($L = 1$ was used in all of the presented calculations). In the case of the trivial method, we calculated the interactions for particles in the central and other cells for all cases, where the center of the image cell was closer to the central one than $100L$. It meant, slightly more than four million image cells for each interaction. We should mention, that a $25L$ or a $50L$ radius for the system supplied satisfactory results. We used $100L$ only to get very accurate data.

We calculated two types of data sets, with random choice of the second particle position for the polynomial and rational function fits. 10,000 random positions (vectors) were stored for the second particle in the first

type, and the following inequalities were applied:

$$0 \leq z \leq y \leq x \leq 0.5 \quad (6)$$

We calculated ten sets of these data. The second type of data sets was designed for the fit process. We intended to fit the potential energy by polynomials and rational functions, but we wanted to use the fitted functions for the calculation of both potential energies and forces. Since, we used only the $1/48$ part of the cell, we had to take care of the curvature of the fitted function at the surface of our elementary tetrahedron as well. The usual concept could be the incorporation of some constraints into the fit, like it is useful in the case of one-dimensional cubic splines and fast interpolation methods [22]. Unfortunately, the situation was rather complicated in our three-dimensional case, and the non-violable constraints might reduce remarkably the degree of freedom in the polynomials. Therefore, we applied a different method to ensure smooth behavior at the surfaces of the tetrahedron. We calculated data sets, where we allowed slight violence of the inequalities $0 \leq z \leq y \leq x \leq 0.5$. We performed it by adding a random number of the $[-0.05; 0.05]$ interval to the non-violating coordinates. Geometrically, it meant an enlargement of the tetrahedron in all directions. We calculated two sets of 10,000 particle coordinates with this second type of method. These sets were used only in the parameterization procedures, while the first type of coordinates was used only in the test of the fitted functions.

We omitted the intra term ($\approx r_{ij}^{11}$ within the central cell) in the fit, because it can be calculated easily in an explicit way. We defined an interaction called as a correction term:

$$\begin{aligned} E_{ij}^{\text{corr}} &= z_i z_j \sum_{l \neq 1}^N 1/r_{ii}^{1l} + z_i z_j \sum_{l \neq 1}^N \sum_{j \neq i}^2 1/r_{ij}^{1l} \\ &= \sum_{l \neq 1}^N 1/r_{ii}^{1l} - \sum_{l \neq 1}^N \sum_{j \neq i}^2 1/r_{ij}^{1l} \end{aligned} \quad (7)$$

The intuitive description of the term, that it is the interaction between two particles: particle one in the central cell and particle one and two in the image cells, if $n = 2$. The charges are omitted on the right hand side, because i and j were oppositely charged with unit absolute value in our two-particle systems. The total potential energy of one cell with arbitrary n can be calculated in a pair-wise manner, in accordance with the previous section about the partition of the energy in periodic systems. Supposing that, the net charge is zero in the central cell:

$$\begin{aligned} U^{\text{tot}}/N &= \frac{1}{2} \sum_{i=1}^n \sum_{j \neq i}^n z_i z_j \left(1/r_{ij}^{11} - E_{ij}^{\text{corr}} \right) \\ &= \frac{1}{2} \sum_{i=1}^n \sum_{j \neq i}^n \left(u_{ij}^{11} - z_i z_j E_{ij}^{\text{corr}} \right) \end{aligned} \quad (8)$$

The net zero charge is a necessary condition, otherwise U^{tot}/N contains incorrect number of self term interactions.

The coefficients of the polynomials were determined in the usual way of ordinary least square fits on $E_j^{\text{corr}}(x_j, y_j, z_j)$. The index of the first particle (i -th particle) was omitted, because this particle was in the origin of the two-particle periodic systems. The polynomials contained all possible terms up to a defined power:

$$P_M(x, y, z) = \sum^{\text{per}} a_{bcd} x^b y^c z^d \quad (9)$$

The superscript per denotes all the possible terms with different b , c and d , where b , c and d are integers of $[0; M]$, M is the highest power of the polynomial, and $(b + c + d) \leq M$. We fitted seven polynomials, $1 \leq M \leq 7$.

In the case of the rational function fit, we used $R_{MS}(x, y, z) = (Q_M(x, y, z))/(Q_S(x, y, z))$, where the numerator and the denominator were similar polynomials as the ones defined in equation (9), but $5 \leq M + S \leq 7$. The optimal coefficients were determined by the method of Gauss–Newton–Marquardt [22] as it is implemented in the Mathematica software [23]. In the test calculations, the potential energies were calculated according to the fitted functions. The forces were calculated as the corresponding partial derivatives of the fitted functions.

In the case of the tabulated calculations, we defined a grid inside the tetrahedron described above. We used the $1/48$ reduction of the central cell to spare the computational memory. The tetrahedron was enlarged to have at least two grid points in all the three Cartesian directions around the non-violating tetrahedron. We calculated the energies and also the x , y and z forces at all grid points. The intra term was subtracted from the data to be consistent to the polynomial and rational function fits. Since, a cell with unit edge lengths was used, a bin size $\Delta r = 0.02$ meant $25 + 4$ grid points in all directions. It meant $29 \times 30 \times 31/6 = 4495$ grid points, $\Delta r = 0.01$ with 54 grids/direction counted 27,720 points, and $\Delta r = 0.005$ with 104 grids/direction gave 192,920 data points. We applied two interpolation formulas in the test calculations. A tri-linear form [23] was applied as

$$E_l^{\text{corr}} = \sum_{i=1}^{I+1} \sum_{j=J}^{J+1} \sum_{k=K}^{K+1} (1 - |x - x_i|)(1 - |y - y_j|) \times (1 - |z - z_k|) E_{ijk}^{\text{corr}} / \Delta r^3 \quad (10)$$

x laid between x_I and x_{I+1} , y laid between y_J and y_{J+1} , z laid between z_K and z_{K+1} .

The other interpolation method was a tri-cubic one [22]. The interpolation was performed as one-dimensional cubic polynomial interpolation at first in the x , then in the y , and then in the z directions. One-dimensional cubic polynomials were fitted at first through the $\mathbf{r}_{I-1,j,k}$, $\mathbf{r}_{I,j,k}$, $\mathbf{r}_{I+1,j,k}$ and $\mathbf{r}_{I+2,j,k}$ points with $J-1 \leq j \leq J+2$ and $K-1 \leq k \leq K+2$, and the function values were interpolated to the $\mathbf{r}_{x,j,k}$ points. Thereafter it was repeated for the $\mathbf{r}_{x,J-1,k}$, $\mathbf{r}_{x,J,k}$, $\mathbf{r}_{x,J+1,k}$ and $\mathbf{r}_{x,J+2,k}$ points with $K-1 \leq k \leq K+2$, and the function values were interpolated to the $\mathbf{r}_{x,y,k}$ points. Finally these last points

were fitted by a cubic polynomial and the $\mathbf{r}_{x,y,z}$ point was interpolated. The polynomials were determined via the Lagrange method. It is rather expensive method of calculation, and it is not recommended in time-consuming computer simulations, but as our results showed, the tri-cubic interpolation has very limited advantages anyway.

To compare our data with previous analytic functions fitted, we choose three functions of Adams and Dubey [20]. The first one was an isotropic approximation, where the interaction over the $1/r$ minimal image convention term was described as

$$E_j^{\text{corr}}(x_j, y_j, z_j) = 2.75022r^2 - 2.94414r^4 + 0.86910r^6 \quad (11)$$

The other two equations based on a series extension of cubic harmonics equation (18) in Ref. [20], where the potential energy is expanded as a finite sum of functions depending on r^l and $(x^l + y^l + z^l)$. Adams and Dubey determined different coefficients for sets including all even l powers up to a given maxima of l values. They proposed, theoretically derived and computationally optimized sets up to $l = 20$. Unfortunately, there were typing errors in their tables (see also in Refs. [24, 25]), therefore the $l > 14$ sets did not provide satisfactory results. We used in our comparison the theoretical set with a maximal $l = 14$ and an optimized set with maximal $l = 12$.

4. Results and discussions

The results of the test calculations on the 10 times 10,000 data points are summarized in tables 1–3. Table 1 contains the data on the fit of the potential to compare the polynomial fits of up to the 7th degree, the rational function fits, where the sum of the degrees in the numerator and in the denominator were in the range of 5–7, data calculated with the aim of tri-linear and tri-cubic interpolations, and data obtained with previous analytical fits. The merit-function of the fits was defined as the average square difference of the original correction and the fitted one. One can see that, the polynomial of the 7th degree provided the best results, and the second best one was obtained by a rational function of 6/1 degrees. The performance of the tabulated methods was rather weak, they were slightly better than a polynomial of the second degree. The isotropic fit of Adams and Dubey, performed weak. The optimized cubic harmonic fit ($l \leq 12$) provided better results, than the cubic harmonic with theoretical parameters with more functions ($l \leq 14$). The optimized set provided accuracy around polynomials of the 4th or 5th degree. The trends of our fits are clear in this column of table 1, the polynomials of high degree performed better. In the case of rational functions, the increasing sum of the degrees was the most important factor. If the sum was equal for two rational functions, the one with higher degree in the numerator provided better data. The tabulated

Table 1. Comparison of potential energy for the polynomial fits, rational function fits, tabulated methods and cubic harmonic fits.

Type		$\langle (E_i^{\text{fit/tab}} - E_i)^2 \rangle$	$100 \langle \left \frac{E_i^{\text{fit/tab}} - E_i}{E_i} \right \rangle$	$100 \langle \left \frac{E_i^{\text{fit/tab}} - E_i}{U_i} \right \rangle$	$\langle E_i^{\text{fit/tab}} - E_i \rangle$	$100 \left \frac{\sum (E_i^{\text{fit/tab}} - E_i)}{\sum E_i} \right $
Polynomial fit	P-1	1.8×10^{-3}	1.5×10^5	1.8×10^0	3.2×10^{-2}	1.2×10^0
	P-2	5.9×10^{-5}	6.7×10^4	2.8×10^1	5.6×10^{-3}	4.6×10^{-1}
	P-3	3.2×10^{-6}	4.2×10^3	6.9×10^{-2}	1.3×10^{-3}	2.3×10^{-2}
	P-4	3.1×10^{-7}	2.3×10^3	2.0×10^{-2}	4.1×10^{-4}	1.3×10^{-2}
	P-5	2.6×10^{-8}	1.7×10^2	5.7×10^{-3}	1.1×10^{-4}	9.6×10^3
	P-6	2.4×10^{-9}	1.1×10^2	1.7×10^{-3}	3.4×10^{-5}	7.3×10^{-4}
	P-7	2.3×10^{-10}	1.1×10^2	5.3×10^{-4}	1.0×10^{-5}	6.2×10^{-4}
Rational function fit	R-3/2	5.9×10^{-7}	4.0×10^3	3.4×10^{-2}	5.7×10^{-4}	2.3×10^{-1}
	R-4/1	1.4×10^{-7}	4.1×10^3	1.3×10^{-2}	2.7×10^{-4}	2.2×10^{-2}
	R-3/3	7.5×10^{-8}	3.7×10^3	1.1×10^{-2}	2.2×10^{-4}	3.6×10^{-2}
	R-4/2	2.4×10^{-8}	1.1×10^3	7.0×10^{-3}	1.3×10^{-4}	4.8×10^{-3}
	R-5/1	1.3×10^{-8}	1.8×10^2	4.3×10^{-3}	8.1×10^{-5}	2.9×10^{-3}
	R-4/3	6.7×10^{-9}	6.3×10^2	3.5×10^{-3}	6.5×10^{-5}	7.6×10^{-3}
	R-5/2	2.2×10^{-9}	1.2×10^2	2.0×10^{-3}	3.6×10^{-5}	2.8×10^{-3}
	R-6/1	8.0×10^{-10}	2.3×10^2	1.0×10^{-3}	2.1×10^{-5}	1.3×10^{-3}
Tabulated tri-linear	Small	6.8×10^{-5}	6.2×10^0	1.2×10^{-1}	1.4×10^{-3}	2.1×10^0
	Medium	3.3×10^{-5}	1.5×10^0	6.0×10^{-2}	7.1×10^{-4}	1.1×10^0
	Large	1.7×10^{-5}	6.5×10^{-1}	3.0×10^{-2}	3.5×10^{-4}	5.3×10^{-1}
Tabulated tri-cubic	Small	4.9×10^{-5}	4.7×10^1	1.7×10^{-1}	2.0×10^{-3}	1.4×10^0
	Medium	2.5×10^{-5}	2.1×10^0	9.1×10^{-2}	1.1×10^{-3}	7.7×10^{-1}
	Large	1.3×10^{-5}	7.0×10^{-1}	4.8×10^{-2}	5.6×10^{-4}	4.0×10^{-1}
Isotr. fit [20]	≤ 6	3.0×10^{-3}	2.6×10^3	2.4×10^0	4.5×10^{-2}	3.3×10^1
Cubic harmonics [20]	≤ 14	3.8×10^{-6}	1.2×10^{-1}	3.5×10^{-2}	3.1×10^{-4}	1.9×10^{-1}
	$\leq 12\text{opt.}$	1.1×10^{-7}	2.7×10^0	1.1×10^{-2}	1.7×10^{-4}	1.1×10^{-2}

For statistical and other details see the text.

methods were sensitive primarily to the grid of the tabulation. The tri-cubic interpolation method performed slightly better than the simple tri-linear one.

At the creation of tabulated data, the average square difference was not a criterion as in the parameterization methods. Therefore, it is necessary to use other measures in the comparison of the different methods. The next three columns contain further data. Surprisingly, the tabulated method performed with three magnitudes better, if we calculate the averaged relative difference of the fitted and original corrections: $100 \langle |(E_i^{\text{fit/tab}} - E_i)/E_i| \rangle$. How is this reverse order possible? How can be this relative average error of the fits around and over 100%? We checked the worst data, and we found, that there are some points in each data set, where the two particles are very close to each other. In these cases, the correction terms are very small and the intra part of the energy can be about 10 magnitudes larger than E_i -s. For example, in the worst data of the polynomial fit of 7th degree U_i was around -170 , E_i was -1×10^{-9} , and E_i^{fit} was 7×10^{-5} . It means a 7,000,000% relative error at this point, but in the absolute scale, the error is only around 10^{-5} , or relatively to the intra term, it is only 0.00004%. The polynomial and the rational function fits performed poorly in this relative comparison, while the tabulation works stable. The cubic harmonic methods were better in this comparison, than the polynomials. If we use the total interaction in the normalizations, $100 \langle |(E_i^{\text{fit/tab}} - E_i)/U_i| \rangle$, the fit methods provides about two magnitudes better results than the tabulated ones. If this criterion was used, the worst data represented the cases, where the intra terms were small, the second particles were close to the corner of the cell, and the

correction term was very important. The best average error was 0.0005% in the fits and 0.05% in the tabulation in this comparison. If we simply compared $\langle |E_i^{\text{fit/tab}} - E_i| \rangle$, the fits of high degrees performed one magnitude better than the tabulation. The calculation of potential energy has a crucial role in the Monte Carlo simulation methods. Here, the total energy of the system is important. It means, the fitting and the tabulated methods should work correctly not only for one interaction, but it is necessary to provide good total energies. Each of our test sets contained 10,000 data. The number of these interactions corresponds to a system about 140 charged particles. The last column in table 1 shows, the average in the 10 sets of $100 \left| \frac{\sum (E_i^{\text{fit/tab}} - E_i)}{\sum E_i} \right|$. On can see, the best fits gave the overall corrections with 0.0006% error. The tabulation provides 0.4% error. We concluded, that the polynomial and rational function parameterizations of high degree provided expressively better results than the tabulation. The tabulation was superior only in one relative comparison, but here the defective points of the fits were those ones, where the corrections were very small and unimportant on the absolute scale. The performance of the optimized cubic harmonic fit in the last three comparisons was similarly around the polynomials of 4th and 5th degree.

The two most important methods in the classical mechanical modeling of liquid matter, are the Monte Carlo and the molecular dynamic simulations. In the case of Monte Carlo methods, the calculation of the potential energy is necessary, so the above fits can be important. In the case of molecular dynamics, the calculation of forces is essential. We present the same table for the forces as table 1 for the potential. The forces were calculated

Table 2. Comparison of forces for the polynomial fits, rational function fits, tabulated methods and cubic harmonic fits.

Type		$\langle (F_i^{\text{fit/tab}} - F_i)^2 \rangle$	$100 \langle \left \frac{F_i^{\text{fit/tab}} - F_i}{F_i} \right \rangle$	$100 \langle \left \frac{F_i^{\text{fit/tab}} - F_i}{F_i^{\text{all}}} \right \rangle$	$\langle F_i^{\text{fit/tab}} - F_i \rangle$
Polynomial fit	P-2	1.9×10^{-2}	1.8×10^3	1.5×10^1	9.6×10^{-2}
	P-3	1.5×10^{-3}	2.7×10^2	5.2×10^0	2.4×10^{-2}
	P-4	1.9×10^{-4}	2.6×10^2	1.4×10^0	8.4×10^{-3}
	P-5	1.5×10^{-5}	6.3×10^1	4.6×10^{-1}	2.4×10^{-3}
	P-6	1.4×10^{-6}	1.6×10^1	1.1×10^{-1}	7.7×10^{-4}
	P-7	1.8×10^{-7}	4.1×10^0	3.9×10^{-2}	2.6×10^{-4}
Rational function fit	R-3/2	3.1×10^{-4}	1.1×10^2	2.6×10^0	1.1×10^{-2}
	R-4/1	6.0×10^{-5}	1.9×10^2	8.5×10^{-1}	5.2×10^{-3}
	R-3/3	3.5×10^{-5}	5.5×10^1	9.4×10^{-1}	4.2×10^{-3}
	R-4/2	1.2×10^{-5}	5.9×10^1	4.9×10^{-1}	2.4×10^{-3}
	R-5/1	7.8×10^{-6}	4.0×10^1	3.4×10^{-1}	1.7×10^{-3}
	R-4/3	3.8×10^{-6}	4.3×10^1	2.5×10^{-1}	1.3×10^{-3}
	R-5/2	1.4×10^{-6}	2.3×10^1	1.9×10^{-1}	7.3×10^{-4}
	R-6/1	4.1×10^{-7}	1.6×10^1	6.1×10^{-2}	4.4×10^{-4}
Tabulated tri-linear	Small	7.2×10^{-4}	2.4×10^0	2.7×10^{-1}	4.7×10^{-3}
	Medium	3.5×10^{-4}	9.5×10^{-1}	1.2×10^{-1}	2.3×10^{-3}
	Large	1.8×10^{-4}	5.0×10^{-1}	5.7×10^{-2}	1.1×10^{-3}
Tabulated tri-cubic	Small	5.5×10^{-4}	3.2×10^0	4.0×10^{-1}	6.9×10^{-3}
	Medium	2.7×10^{-4}	1.5×10^0	1.8×10^{-1}	3.6×10^{-3}
	Large	1.4×10^{-4}	6.6×10^{-1}	9.4×10^{-2}	1.8×10^{-3}
Isotr. fit [20]	≤ 6	2.4×10^{-1}	4.7×10^2	1.5×10^1	3.4×10^1
Cubic harmonics [20]	≤ 14	6.5×10^4	9.5×10^1	2.4×10^1	4.7×10^3
	$\leq 12\text{opt.}$	3.0×10^{-5}	1.0×10^0	1.4×10^{-1}	2.5×10^{-3}

For statistical and other details see the text.

simply as the corresponding x , y and z partial derivatives of the fitted potential. In the case of the tabulation, the forces were tabulated similarly to the potential, and the same tri-linear or tri-cubic interpolation methods were applied. F_i denotes the original correction term, $F_i^{\text{fit/tab}}$ is the fitted or tabulated value of the correction term, and F_i^{all} means the sum of the correction and the intra term. The data shown in table 2 are the averages over the ten test sets and the three directions: x , y , and z . Trends were similar to the ones in table 1. The fits with polynomials or rational functions of high degree performed the best. In the case of the relative comparison to the force correction, the tri-linear seems to be superior, but the apparently weak performance of the fits did not mean serious deficiency in this comparison. We checked the worst data, and we concluded similarly as in the case of the potential. The weak average originates from the points, where the contribution to the force from the intra part is about ten magnitudes larger than the correction term. The fits provide maximum about four magnitudes larger corrections, than the real ones, but these ones are still negligible in respect to the total force. We omitted the line, which corresponds to the total force within one data set, because it is not important to calculate it in molecular dynamics. The optimized cubic harmonic sets performed again better than the theoretical sets. Its accuracy laid between the polynomials of 4th and 5th degree for the first comparison, but in the third and fourth comparison they provided slightly better data than the polynomials of 5th degree.

We did not provide confidence intervals for the data in table 1 and 2, because we did not want to overcrowd the tables. We calculated the error intervals for the data with a significance of $\alpha = 0.05$ on the 10 test sets. For the force

data it meant 30 sets taking an average over the x , y , and z directions. The relative uncertainty of the data was slightly different for the various columns in the two tables. The average error was between 0.02 and 0.06% for the first, third and 4th columns of table 1 in the cases of polynomial and rational function fits. Respectively, it was in the 0.05–0.2% range for the force data in table 2. The corresponding values for the tabulated data were 2–5 times larger. In the case of the second columns, the error bars were smaller for the tabulation (for the energy around 1.5 and 0.2% for the forces), and the data for the fits were about three times larger. The uncertainty of the 5th column in table 1 was 0.8% for the fits and 0.2% for the tabulations.

If one looks at table 1 and 2, the optimized solution would be a polynomial fit of high degree. Of course, in the case of computational methods, the requested number of operations is an important factor. We summarized the number of different operations for the calculation of potential energy and force in table 3. The data were calculated in respect to one interaction. The calculation cost of the distance of the two interacting particle is not included in the table. In the case of the polynomial and rational function fits and the tabulation, the calculation cost of the bookkeeping of the original signs of the coordinates and of the rotation of the vector to provide $0 \leq z \leq y \leq x \leq 0.5$ was not counted. The table contains data on the operations demanded by the potential calculation via the Ewald method as it was implemented in the supplement of the book of Allen and Tildesley [11]. If one reports on the scaling of the Ewald method, there are different data from $n-n^2$, both for the direct and the reciprocal space sums. The difference in the data originates from the adjustable width of the Gaussian distribution that

Table 3. Number of instructions for the calculation of one interaction.

Type		Number of instructions	
		Potential	Forces and potential
Polynomial fit	P-1	6	–
	P-2	18	30
	P-3	38	74
	P-4	68	144
	P-5	110	246
	P-6	166	386
	P-7	238	570
Rational function fit	R-3/2	59	120
	R-4/1	77	166
	R-3/3	79	164
	R-4/2	89	190
	R-5/1	119	268
	R-4/3	109	234
	R-5/2	131	292
	R-6/1	175	408
Tabulated tri-linear	Small	215	431
	Medium	215	431
	Large	295	751
Tabulated tri-cubic	Small	2615	6555
	Medium	2615	6555
	Large	3255	9115
Ewald	$n = 10$	710	2724
	$n = 50$	143	514
	$n = 100$	93	278
	$n = 10000$	42	65
Isotr. fit [20]	≤ 6	10	22
Cubic harmonics [20]	≤ 14	145	629
	$\leq 12\text{opt.}$	121	537

The unit of the operations is one multiplication or addition.

is a free parameter in the Ewald method. If the Gaussians are chosen to vanish at a cutoff of half the cell size, the direct sum scales n^2 and the reciprocal sum scales n . If the Gaussian width is chosen to provide Gaussian vanishing at a standard cutoff distance independent of n , the direct sum scales n , but the reciprocal sum becomes n^2 scaling [2,7]. By varying the cutoff with the square root of the cell size, it can be shown that both the direct and reciprocal sum can be scaled $n^{3/2}$, which is supposed to be optimal by some authors [26]. As we mentioned in the introduction, the primary aim of our investigation was to get a practically usable interpretation of the periodic Coulomb interactions in the case of program codes, where the interactions are calculated via one by one. In the general implementation of the Ewald method, the calculation is performed on the interactions via one by one for the direct sum, but the reciprocal space sum is calculated only once for all of the interactions in the system. Therefore, if we would like to compare our methods to the Ewald one, we have to define a system size (n) and also the choice of the Gaussian width. The data in table 3 correspond to a linear scaling in the reciprocal space sum and an n^2 in the direct part. The system sizes are indicated in the table, too. The unit of the number of the operations corresponds to one multiplication or addition. The other operations, as division, calculation of exponentials, trigonometric functions and logic functions were counted in these units. The ratios were obtained in small test calculations. In these tests, we calculated the cost

of the different functions by small programs, where the CPU time of a million of these calculations were counted relatively to a million of multiplications or additions. The codes were written in Fortran and C, and they were used on PC-s with Pentium IV processor running under Windows XP. We found, no relevant differences in the ratios in the cases of the two programming languages. Maybe it is a consequence of that we used the compilers from the same vendor for the two languages. There were not any significant differences in the calculation cost of a double precision addition and multiplication. An exponential took 22.9, a sine or cosine took 11.4, a division took 2.6 and a conditional jump took 1.6 times longer time, than a multiplication, addition or subtraction. The truncation of a real number to an integer cost 1.6 units. A search in a vector depended on the size, for vectors up to a few ten thousands elements it was three times more expensive than a multiplication, but it increased up to 13 times more for a vector with 1,92,920 elements.

The corresponding column of table 3 contains the multiplication equivalent calculation cost for the different methods. In the case of polynomials, the calculation needs solely multiplications and additions in a half–half ratio, if one uses the computationally feasible Horner arrangements. In the case of the rational functions, both the numerator and the denominator can be expressed in the Horner form. It was rather surprising, that the number of the operations were rather high already for the simple tri-linear interpolation. In the case of the tri-cubic interpolation, the operational cost was extremely high. It was coded with the expensive one-dimensional Lagrange method, but significant reduction is not possible with any other cubic methods.

If one compares the data of table 1 and 3, we can find optimal solutions for the potential calculations. Let us compare the polynomial, the rational function, the tabulated methods and the cubic harmonics at first. The tabulated ones performed weakly with respect to the computational cost. Therefore, we do not suggest the use tabulated methods. It is not easy to choose from the polynomials and rational functions of different degrees. Depending on the desired accuracy, we suggest the following order: $P-3 < R-3/2 < P-4 < R-3/3 < R-4/3 < R-5/2 < R-6/1 < P-7$. The notations correspond to the acronyms in the tables. We omitted some of the fits, where other fits with similar computational cost were close to them with higher accuracy. The number of the operations with respect to the merit function is visualized for the parameterized functions in figure 3. The computational cost of the optimized cubic harmonics ($l \leq 12$) was slightly larger than the polynomial of 5th degree, but relevant difference was not found. In comparison with the rational functions the use of cubic harmonics seemed to be less feasible.

If one compares the performance of the fits to the Ewald method, one can find optimal solutions with respect to the system size. A polynomial of the third degree is less expensive computationally than the Ewald method at all

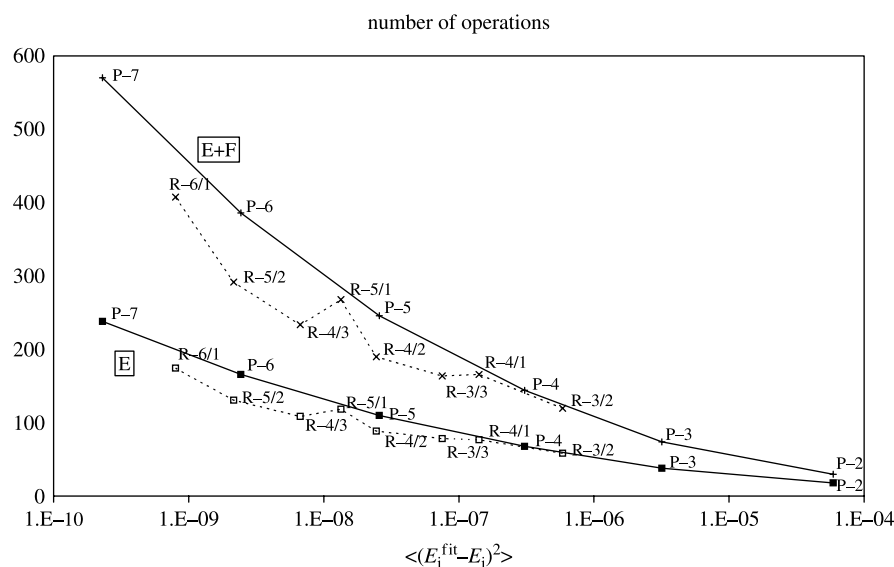


Figure 3. Number of operations with respect to the merit function of the fits (second column of table I). Energy calculations: \square polynomials, \blacksquare rational functions. Energy and force calculations: $+$ polynomials, \times rational functions. The data are connected with solid and dashed lines to clarify trends.

system sizes. An accurate fit costs more than the Ewald method for large systems. At the system size around 100 particles, or more explicitly, where the number of the charged particles is around 100, the rational functions with a degree of 4/3 or 3/3 are computationally comparable to the Ewald method. For small systems (less than 50 charged particles) the use of whatever fit is feasible.

The situation is different, if one is interested in the forces, too. Here one has to calculate 4 values: the x , y and z components of the forces and the potential. The estimated total computational costs of these calculations are summarized in table 3. The data corresponding to the fits are approximate. There are many common terms in the Horner expressions of the potential and x , y , and z components of the forces in the polynomials and rational functions. Furthermore, the degree of the functions is one less in the force calculation than in the potential calculation. An effective compiler may omit the redundant calculations. We guessed that the cost of these calculations might be reduced to the calculation of two forces and the potential instead of three forces and the potential. The data in table 3 include this reduction also for the cubic harmonics. As one can see in table 3, the force calculation with the cubic harmonics is expensive. The used degree of r , x , y , and z is rather large in the cubic harmonic expansion, and the loss of degree is less significant in comparison to the rational functions or polynomials. The calculation of a cubic harmonic force component cost more than a calculation of a potential value, while the opposite case happened for our polynomials and rational functions. The calculation cost of the optimized $l \leq 12$ cubic harmonics is around a polynomial with a degree of 7th, but the accuracy of the latter is reasonably better. The same accuracy can be achieved with a half of the computation time with polynomials and especially with rational functions, if both potential energy and forces are calculated. Therefore, it is feasible to use our new functions especially in

molecular dynamics and in other applications, where the forces are important as well. The common calculation of forces and potential slightly modified the preferences in respect to the Ewald method. The polynomial of third degree became slightly less effective, than the Ewald method for large systems. On contrary, for systems consisting about 100 charged molecules one can efficiently use more precise polynomials than in the case of simple potential calculations. The rational function 6/1 fit seemed to be the best choice for systems consisting of less than 50 particles (see also in figure 3).

5. Conclusions

The aim of our study, was to elaborate a simple method to calculate Coulomb interactions in three-dimensional periodic systems. The method is not intended to be competitive with well established methods on physical intuition. We would like to present a feasible way to calculate the long-range interactions in program codes, primarily designed for systems with short-range interactions. We have shown that, the crucial part of the Coulomb interaction, which is the extra term besides the contribution within the minimal image convention, can be easily parameterized with simple functions. We used polynomials of up to the 7th degree and rational functions with a sum of 5, 6 and 7 of the degrees in the numerators and denominators. The comparison of the fits to simple tabulated functions showed that, the accuracy of the fits is reasonably better than for the tabulations, especially if we took into account the computational cost of the methods. Our functions seem to be as feasible as the previous cubic harmonic parameterizations, if solely the potential energy is calculated. The comparison was performed both for simple potential calculation and for common calculation of forces and potentials. The first corresponded to the

Monte Carlo simulations, while the second concerned the requirements of molecular dynamics. If both potential energy and forces are needed, the polynomials and rational functions provide the same accuracy with a half of computational cost than it is required for cubic harmonics. We proposed some sets of parameterized functions. The suggested functions are detailed in the Appendix.

Of course, our parameterized functions should be competitive not only during the program code development, but they should not mean a drastic increase in the computational cost during the simulations. Therefore, we compared the calculation cost of our methods to the Ewald one. We obtained, reasonable good performance for our methods in the case of small systems. In the case of medium and large systems, our parameterized polynomials and rational functions and the previously fitted cubic harmonics are competitive in the calculation cost, only if one uses less accurate fits.

The methods are intended to be easily embeddable into existing codes. One needs to add them as a function into the codes and call them at each interaction pair.

Acknowledgements

The authors thank the tenure of the grant Hungarian Research Fund T43542. The discussions with Dr Z. Bajnok are also acknowledged.

Appendix

Selected parameterized functions are detailed here. The other functions can be accessed at the homepage of the authors*. The forces can be calculated as the corresponding partial derivatives of the functions. The format of the functions helps to paste them directly into C or Fortran program codes.

P-7 polynomial

$$E^{\text{corr}}(x, y, z) = 0.00006935528245746797 + z*(0.0021600087427982575 + z*(0.010077826502256547 + z*(0.08211063320794686 + z*(-3.4732828079283466 + z*(-9.314273611252304 + (14.16681370479069 - 3.111726172826123*z)*z)))) + y*(0.001575589085250862 + z*(-0.04946302738211095 + z*(-0.5461572397611817 + z*(9.645235058499194 + z*(-12.100970067330422 + z*(10.468290488143378 + 6.309005153979316*z)))) + y*(-0.12396063682007008 + y*(0.982231868071471 + z*(6.028068019846833 + z*(-17.789466989160736 + (-96.5199335183439 - 42.32265420041536*z)*z)) + y*(-0.9283843203267544 + y*(-27.812214458566334 + y*(21.30693743338091 + 4.520106137117244 0106137117244*y + 5.422380955735728*z) + z*(-33.42332128349194 + 10.254096587054391*z)) + z*(-9.124340828920644 + z*(98.53115658291732 + 30.21428274012793*z)))) + z*(0.469739555622663 + z*(9.73056725774623 + z*(-19.326699784455506 + z*(-11.37420631739499 + 59.2737824517607*z)))) + x*(-0.0010769741023264508 + z*(-0.03331222484387889 + z*(0.12239075346337186 + z*(-8.464141594766453 + z*(32.13297640506034 + (5.295849771329786 - 26.680059631432933*z)*z)))) + y*(0.06432376081037171 + z*(0.48142207166094036 + z*(-3.456674703281202 + z*(-29.470389357353614 + (63.421756702461884 - 141.98270528757035*z)*z)))) + y*(0.5714542131297252 + z*(-21.095935501594877 + z*(60.85631110259151 + (363.4053322178775 - 3.385620913504718*z)*z)) + y*(-19.141586520349748 + y*(67.1392836634568 + (151.95009370062058 - 296.24604740512046*z)*z + y*(77.75706782621464 - 81.30021189598561*y + 50.407253193861436*z)) + z*(-22.584361540660517 + z*(-94.37317073592516 + 109.45214658704292*z)))) + x*(-0.006065872558772311 + z*(0.30113981770468445 + z*(13.312076580467963 + z*(29.617911396911396 + z*(-116.72457001390151 + 107.43007317129914*z)))) + y*(-0.6255143663443122 + z*(10.524061781990161 + z*(-8.57249448295761 + z*(-273.58407474601285 + 77.40904240322111*z)))) + y*(21.83680155975584 + z*(123.72400023547327 + (-333.2887060666331 - 582.325447006672*z)*z) + y*(21.326945569638394 + y*(-334.6661821372041 + 80.44298584353629*y - 249.48921174203804*z) + z*(-131.7347948910907 + 542.7920598545852*z)))) + x*(0.047081255188474065 + z*(-4.043834137787724 + z*(-21.28091606308152 + (66.20110806516355 - 11.71291472658606*z)*z)) + x*(-2.634940113127114 + z*(23.224190523041237 + (-9.4717876144749 - 184.09034470257453*z)*z) + y*(26.71884823132592 + (187.66033464664753 - 385.5182685340148*z)*z + y*(54.07626820719544 - 384.66964141880385*y + 7.403361674992371*z)) + x*(-4.328124199512774 + y*(-65.56910963194828 + 86.47299097623085*y - 119.6071622023683*z) + x*(11.549646843718113 - 13.071896023932819*x + 48.37625921900341*y + 37.23947113930151*z) + z*(-50.97079346331456 + 109.92078183137382*z)) + y*(-2.0733112680607295 + y*(-65.23997175235587 + z*(-171.0130695529695 + 125.31168108618317*z))$$

*<http://www.chem.elte.hu>

R-6/1 polynomial:

$$\begin{aligned}
E^{\text{corr}}(x, y, z) = & (-0.0000960512168521368 + z*(-0.005607048507666448 + z*(0.0906226345763506 \\
& + z*(-0.21639277687019992 + z*(-3.277365176264637 + (2.004882272381631 - 0.6512401928726239*z)*z)))) \\
& + y*(-0.007829400157423769 + y*(0.13798121954712936 + z*(-0.8126272788330875 \\
& + z*(11.060280733203957 + (4.441358500803813 - 2.375135450021934*z)*z)) + y*(-0.008639986515872387 \\
& + z*(0.4009488000594831 + (4.270529310803303 - 19.335723163875056*z)*z) + y*(-3.419015489617705 \\
& + y*(-4.0272766940210065 + 0.8304169575975706*y - 2.095211263815416*z) + z*(6.624728386093039 \\
& + 1.242354506552586*z)))) + z*(0.1495830902516656 + z*(-0.8900349273909299 + z*(-1.3091761203010592 \\
& + z*(-0.4933802467002245 + 1.8779096673018691*z)))) + x*(0.004196154599397339 \\
& + z*(-0.00701069724990378 + z*(-0.6943869185678819 + z*(4.669182014814679 + z*(3.633879338225599 \\
& + 1.1393052601669498*z)))) + y*(-0.003933806714890548 + z*(-1.0983974530865155 \\
& + z*(7.788585776402839 + (-2.7875144215781593 + 0.15939476651879092*z)*z)) \\
& + y*(-1.7678580265053072 + y*(1.5481102263919486 + y*(13.37660017591304 \\
& + 13.060615130380311*y - 7.923680438753291*z) + z*(-17.891239477025035 + 15.87717897146121*z)) \\
& + z*(6.009037922746548 + z*(-41.59808673171331 + 21.395376236167262*z)))) \\
& + x*(-0.033272445690500634 + z*(0.584020424351269 + z*(6.523442379770122 + (-12.05263249562082 \\
& - 4.290313716049*z)*z)) + x*(-0.007809226559581139 + x*(-1.580463496237746 \\
& + x*(0.894523744996066 + 2.3599451952315995*x - 12.47838559940495*y - 4.2078185053570625*z) \\
& + (6.090355759865185 - 6.459937570553608*z)*z + y*(13.375904119024488 + 12.785165253609035*y \\
& + 4.832761581301174*z)) + z*(-3.0230947417126575 + z*(-2.5362219185889225 + 6.203539763948587*z)) \\
& + y*(-6.374217731434383 + y*(-23.717360459500625 + 13.23809107886893*y - 12.898330486490876*z) \\
& + z*(-3.7235104862966772 + 17.110057606860188*z)) + y*(1.0263819882673804 + z*(2.169543574904084 \\
& + (-11.3101826530877 + 0.501075586402872*z)*z) + y*(14.238540869774098 + z*(-1.8873012512884682 \\
& + 4.278290557414104*z) + y*(-1.3843737175442605 - 34.26326140949819*y + 29.92194171904781*z))))/ \\
& (0.7785801748034736 - 1.1780056799421226*x + 0.47467769284219097*y - 0.03459234405920862*z)
\end{aligned}$$

R-4/3 polynomial:

$$\begin{aligned}
E^{\text{corr}}(x, y, z) = & (0.276603502858369 + z*(-4.561654391176089 + z*(-16.848531578909338 \\
& + (-496.30931123018576 - 330.4895352528282*z)*z)) + x*(-10.889480971076306 \\
& + y*(-127.82716466691214 + y*(1417.2490046012413 + 610.8986719429023*y - 250.0357830465296*z) \\
& + (-613.048742546201 - 2514.4374426553577*z)*z) + z*(-1.8223169952808964 \\
& + z*(807.2550480845954 + 1081.9222907760686*z)) + x*(127.33552277440984 \\
& + x*(-691.0342370897528 + 531.4267614254303*x + 130.68301042391113*y - 538.1623155961747*z) \\
& + z*(229.83556002084737 + 39.590231961430064*z) + y*(303.4943899078969 - 1570.0668499481665*y \\
& + 1175.7251203585736*z)) + y*(9.597851468039439 + z*(56.881627373512124 + z*(433.15875573482634 \\
& + 704.4361965120456*z)) + y*(-42.95108114953812 + y*(-567.0114168090131 - 62.94770861897754*y \\
& + 52.151196418409896*z) + z*(143.00363221889194 + 1626.282611553713*z))))/(504.83457101003654 \\
& + z*(-246.65352293075762 + (70.6749845912373 - 156.47732101930012*z)*z) + y*(-256.846382918661 \\
& + z*(658.3031284197873 + 180.52973086841993*z) + y*(298.8238930832602 - 489.4813036276625*y \\
& + 269.0487436607461*z)) + x*(-1223.2416233280578 + y*(120.44506592295635 + 514.6341617498665*y \\
& - 1534.3392088908931*z) + x*(1172.607406289951 - 518.2452442689786*x + 124.34877912152452*y \\
& - 664.3639329999492*z) + z*(953.9993392115671 + 442.61872952630995*z))
\end{aligned}$$

R-3/3 polynomial:

$$\begin{aligned}
E^{\text{corr}}(x, y, z) = & (-5.6468521708440065 + z*(-0.6455850054332947 + (-230.05350161010912 \\
& - 54.17650850083276*z)*z) + y*(7.992951149397931 + y*(-87.49761949669715 - 1949.205115702377*y \\
& + 254.24445550679738*z) + z*(181.25043636088205 + 1407.8975806293345*z)) + x*(82.85111615402805 \\
& + y*(-130.87697422086134 + 3887.4574763651417*y - 752.741678587582*z) + x*(-295.7807664461502 \\
& - 621.19046643253*x + 375.94415623137803*y + 232.01135342400596*z) + z*(-66.94866662350915 \\
& + 2827.9582789815113*z))) / (2144.1865380721265 + y*(-3105.804366519403 + y*(-378.9440833358306 \\
& - 1362.783553384834*y - 806.3630159591904*z) + (-1028.4360620641837 - 1779.4015301222191*z)*z) \\
& + z*(526.7280804765501 + (1974.1832548064363 - 296.2714329249807*z)*z) + x*(-5177.023448684519 \\
& + x*(4330.109316156619 - 848.1190906080768*x - 9626.709942481264*y - 897.9320682272385*z) \\
& + z*(-607.5401961600213 + 558.5610651538227*z) + y*(10484.299117231063 \\
& + 3700.5309423965664*y + 2417.052553088499*z))
\end{aligned}$$

P-3 polynomial:

$$\begin{aligned}
E^{\text{corr}}(x, y, z) = & -0.002662936871531699 + z*(-0.03895811249804262 + (0.10436444596789068 \\
& - 2.4819620219519227*z)*z) + y*(0.1216783487028532 + z*(0.8951353654415654 + 1.5726746317550928*z) \\
& + y*(0.6405532694957459 - 3.158266234200411*y + 1.8729798816149168*z)) \\
& + x*(-0.028001857078216615 + y*(-2.668489424206896 + 4.37373697972929*y - 4.873877687390871*z) \\
& + x*(1.107379412223926 - 3.965579695813902*x + 5.927493669743001*y + 3.231343548491743*z) \\
& + z*(-0.9380907165155405 + 4.647533829714995*z))
\end{aligned}$$

References

- [1] E.E. Ewald. *Ann. Phys.*, **64**, 253 (1921).
- [2] C. Sagui, T.A. Darden. *Ann. Rev. Biophys. Biomol. Struct.*, **28**, 155 (1999).
- [3] H.L. Friedman. *Mol. Phys.*, **29**, 1533 (1975).
- [4] M. Neumann, O. Steinhauser. *Mol. Phys.*, **39**, 437 (1980).
- [5] J.W. Eastwood, R.W. Hockney, D. Lawrence. *Comput. Phys. Commun.*, **19**, 215 (1980).
- [6] T.A. Darden, D. York, L. Pedersen. *J. Chem. Phys.*, **98**, 10089 (1993).
- [7] U. Essmann, L. Perera, M.L. Berkowitz, T.A. Darden, H. Lee, L. Pedersen. *J. Chem. Phys.*, **103**, 8577 (1995).
- [8] A.J.C. Ladd. *Mol. Phys.*, **33**, 1039 (1977).
- [9] A.J.C. Ladd. *Mol. Phys.*, **36**, 463 (1978).
- [10] L. Greengard, V. Rokhlin. *J. Comp. Phys.*, **73**, 325 (1987).
- [11] M.P. Allen, D.J. Tildesley. *Computer Simulation of Liquids*, Clarendon, Oxford (1987).
- [12] D.C. Rapaport. *The Art of Molecular Dynamics Simulation*, Cambridge University Press, Cambridge (1995).
- [13] D. Frenkel, B. Smit. *Understanding Molecular Simulation*, Academic, London (1996).
- [14] K. Esselink. *Comp. Phys. Comm.*, **87**, 375 (1995).
- [15] B.A. Luty, I.G. Tironi, W.E. van Gunsteren. *J. Chem. Phys.*, **103**, 3014 (1995).
- [16] E.L. Pollock, J. Glosli. *Comp. Phys. Comm.*, **95**, 93 (1996).
- [17] S.G. Brush, H.L. Sahlin, E. Teller. *J. Chem. Phys.*, **45**, 2102 (1966).
- [18] J.P. Hansen. *Phys. Rev.*, **A 8**, 3096 (1973).
- [19] W.L. Slattery, G.D. Doolen, H.E. DeWitt. *Phys. Rev.*, **A 21**, 2087 (1980).
- [20] D.J. Adams, G.S. Dubey. *J. Comp. Phys.*, **72**, 156 (1987).
- [21] F.G. Anderson, C-H. Leung, F.S. Ham. *J. Phys. A: Math. Gen.*, **33**, 6889 (2000).
- [22] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery. *Numerical Recipes in Fortran*, Cambridge University Press, Cambridge (1992).
- [23] S. Wolfram. *The Mathematica Book*, 4th ed., Wolfram Media / Cambridge University Press, Cambridge (1999).
- [24] I. Ruff, A. Baranyai, E. Spohr, K. Heinzinger. *J. Chem. Phys.*, **91**, 3148 (1989).
- [25] I. Ruff, D.J. Diestler. *J. Chem. Phys.*, **93**, 2032 (1990).
- [26] J.W. Perram, H.G. Petersen, S.W. DeLeeuw. *Mol. Phys.*, **65**, 875 (1988).