

## Fast Computation of Electrostatic Interactions for a Charged Polymer with Applied Field

Lin Hao, Lei Zi-Tong, Ding Ming-Ming, Wang Hong-Jun, Shi Tong-Fei

Cite this article as:

Lin Hao, Lei Zi-Tong, Ding Ming-Ming, Wang Hong-Jun, Shi Tong-Fei. Fast Computation of Electrostatic Interactions for a Charged Polymer with Applied Field[J]. *Chinese J. Polym. Sci.*, 2020, 38(6): 665-672. doi: 10.1007/s10118-020-2343-8

View online: <https://doi.org/10.1007/s10118-020-2343-8>

---

### Articles you may be interested in

Numerical Study of the Interplay of Monomer–Surface Electrostatic and Non–electrostatic Interactions in the Adsorption of Weak Polyelectrolytes on Oppositely Charged Surfaces

Chinese J. Polym. Sci. 2016, 34(5): 552 <https://doi.org/10.1007/s10118-016-1780-x>

INTERACTION OF SIMILARLY CHARGED SURFACES MEDIATED BY NANOPARTICLES

Chinese J. Polym. Sci. 2011, 29(4): 414 <https://doi.org/10.1007/s10118-011-1053-7>

Temperature–responsive Behavior of Polymer Fluorescent System *via* Electrostatic Interaction Mediated Aggregation/Deaggregation

Chinese J. Polym. Sci. 2016, 34(7): 830 <https://doi.org/10.1007/s10118-016-1793-5>

PS纳米球在羧甲基壳聚糖自组装膜表面静电吸附行为研究

ELECTROSTATIC ADSORPTION OF SURFACE CHARGED POLYSTYRENE NANOSPHERES ON CARBOXYMETHYLATED CHITOSAN SELF ASSEMBLED MONOLAYER

高分子学报. 2010(8): 1050 <https://doi.org/10.3724/SP.J.1105.2010.09419>

疏水缔合和静电作用调控P(DTAB-co-AM)与蠕虫状胶束自组装网络

SELF-ASSEMBLY NETWORKS OF P(DTAB-co-AM) AND WORMLIKE MICELLES MEDIATED BY HYDROPHOBIC ASSOCIATION AND ELECTROSTATIC INTERACTIONS

高分子学报. 2009(9): 879 <https://doi.org/10.3724/SP.J.1105.2009.00879>

水溶液中管状带电纳米颗粒跨膜运输的分子动力学模拟

MOLECULAR DYNAMICS SIMULATION OF THE TRANSPORT OF CHARGED TUBE-LIKE NANOPARTICLES THROUGH A FLUIDIC CHANNEL

高分子学报. 2013(12): 1561 <https://doi.org/10.3724/SP.J.1105.2013.13144>

# Fast Computation of Electrostatic Interactions for a Charged Polymer with Applied Field

Hao Lin<sup>a,b</sup>, Zi-Tong Lei<sup>a</sup>, Ming-Ming Ding<sup>b\*</sup>, Hong-Jun Wang<sup>c\*</sup>, and Tong-Fei Shi<sup>b,d\*</sup>

<sup>a</sup> School of Mathematics, Jilin Normal University, Changchun 130103, China

<sup>b</sup> State Key Laboratory of Polymer Physics and Chemistry, Changchun Institute of Applied Chemistry, Chinese Academy of Sciences, Changchun 130022, China

<sup>c</sup> College of Computer, Jilin Normal University, Siping 136000, China

<sup>d</sup> School of Applied Chemistry and Engineering, University of Science and Technology of China, Hefei 230026, China

**Abstract** Using a hybrid simulation approach that combines a finite difference method with a Brownian dynamics, we investigated the motion of charged polymers. Owing to the fact that polymer-solution systems often contain a large number of particles and the charged polymer chains are in a state of random motion, it is a time-consuming task to calculate the electrostatic interaction of the system. Accordingly, we propose a new strategy to shorten the CPU time by reducing the iteration area. Our simulation results illustrate the effect of preset parameters on CPU time and accuracy, and demonstrate the feasibility of the “local iteration” method. Importantly, we find that the increase in the number of charged beads has no significant influence on the time of global iterations and local iterations. For a number of  $80 \times 80 \times 80$  grids, when the relative error is controlled below 1.5%, the computational efficiency is increased by 8.7 times in the case that contains 500 charged beads. In addition, for a number of  $100 \times 100 \times 100$  grids with 100 charged beads, the computational efficiency can be increased up to 12 times. Our work provides new insights for the optimization of iterative algorithms in special problems.

**Keywords** Finite difference method; Brownian dynamics; Charged polymers; Electrostatic interaction; Local iteration

**Citation:** Lin, H.; Lei, Z. T.; Ding, M. M.; Wang, H. J.; Shi, T. F. Fast computation of electrostatic interactions for a charged polymer with applied field. *Chinese J. Polym. Sci.* 2020, 38, 665–672.

## INTRODUCTION

The diffusion of charged polymers is often the rate-determining step in many biological processes. Stochastic methods (such as Brownian dynamics) and continuum methods (such as finite difference method) are two main computational methods for studying diffusion.<sup>[1–6]</sup> In the process of diffusion, electrostatic force plays an important role in polymer interactions. At present, electrostatic properties of polymers have been extensively studied in the interdisciplinary areas of biomolecule, computational mathematics, physics, and chemistry. There are many methods for calculating electrostatic interactions. However, the existing methods (such as particle-mesh Ewald (PME), Ewald summation, particle-particle particle-mesh (P3M), Coulomb force formula, etc.) all require a lot of CPU time and use large computer memory.<sup>[7–11]</sup> When a huge three-dimensional simulation system is taken into account, it is unsuitable to use these methods to calculate electrostatic forces owing to the serious challenge in computing time. Therefore, an efficient method is urgently needed for saving computing time. The numerical method

is more advantageous, because it sacrifices some accuracy to get a great increase in the calculation speed. At the same time, the numerical method has a weaker dependence on computer memory.<sup>[12]</sup>

Hence, through combining the advantages of traditional methods and numerical methods to simulate the motion of charged polymers is an effective and accurate way, where the numerical methods is mainly used to handle the electrostatic interaction of charged polymers by solving the Poisson equation.<sup>[13]</sup> This is important because many diffusing molecules in biological systems are charged. The key to solving Poisson equation is how to figure out the potential distribution of electrostatic field quickly and accurately by combining the boundary conditions, and provide accurate force analysis for the subsequent simulation of charged polymer motion. Traditional numerical simulations of three-dimensional problems tend to be computationally intensive because of the requirements on the memory and CPU time to obtain solution with desirable accuracy. Even the most advanced computers may not be able to handle them directly. Thus, in electrostatics, calculating the potential distribution of a given charge density function is crucial. In solving such three-dimensional problems, finite difference method (FDM) is a relatively simple and convenient method in many numerical methods, which has the mature theory and the optional accuracy.<sup>[14]</sup> When dealing with the regular area, it is easier to program and im-

\* Corresponding authors, E-mail: mmding@ciac.ac.cn (M.M.D.)

E-mail: jlnwhj@sina.com (H.J.W.)

E-mail: tfshi@ciac.ac.cn (T.F.S.)

Received July 30, 2019; Accepted August 21, 2019; Published online October 11, 2019

plement on computer than the finite element method (FEM) and the finite volume method (FVM).<sup>[15]</sup> The principle of FDM is to discretize Poisson's equation.<sup>[16–20]</sup> Most of the discretized Poisson equations will give a series of linear equations with a large number of unknowns and diagonally dominant coefficient matrices. By using Jacobi, Gauss-Seidel, successive over relaxation (SOR), or Multigrid iterative methods, however, the unknown solutions at the corresponding grid nodes can be approximated iteratively.

Since the iterative algorithm takes up too much CPU time, almost all optimizations are for iteration algorithms. Kamboh *et al.* have proposed the parallel Jacobi method (PJ), which can reduce the computing time for large data moderately.<sup>[21–23]</sup> It is superior to the sequential Jacobi (SJ) and the sequential Gauss-Seidel (SGS). Scheduling Relaxation Jacobian (SRJ) is another method for classical optimization, which greatly speeds up convergence and reduces computation time, and it has certain advantages in dealing with complex boundaries.<sup>[24–31]</sup> In addition, there are also some studies on Multigrid method, for solving 3D Poisson equation with high accuracy, which bases on Richardson extrapolation method.<sup>[32–36]</sup> The fourth-order compact difference discretization scheme and V-cycle algorithm of multigrid are used to solve the Dirichlet boundary value problem of 3D Poisson equation. Finally, the expensive computation of 3D Poisson equation is solved very well. This method is characterized by high accuracy and some speed improvement.

Nonetheless, in Brownian dynamics, we need to calculate millions or even tens of millions of time steps, which means that the calculation of electrostatic interactions will take substantial time even the aforementioned methods are fast in calculation. Therefore, in the process of simulating the movement of charged polymer chains, it is necessary for us to find a new way to save time and ensure accuracy for millions or even more calculations.

In this study, we present “local iteration” to accelerate the simulation of charged polymers's motion, which possesses the irrelevant local iteration process and global iteration process. The simulation model and corresponding optimization algorithm are presented in the second section. And in the third section, we systematically discuss the effect of the parameters in “local iteration” on the simulation efficiency and accuracy, which further verifies the feasibility of our optimization algorithm. Finally, comprehensive summarization of the processes and results of the simulation test are presented in the last section, as well as some prospections.

## SIMULATION MODEL AND METHODS

As shown in Fig. 1, in a simulation box, we fix the left face of electric potential (blue) as  $U$  and the others (gray) as 0, and put into a polymer, which is modelled as a conventional bead-spring chain composed of  $N$  successive Lennard-Jones (LJ)<sup>[37–39]</sup> beads and connected through the finitely extensible nonlinear elastic (FENE) potential.<sup>[40]</sup>

$$U_{\text{LJ}}(r) = 4\epsilon[(\sigma/r)^{12} - (\sigma/r)^6] + \epsilon, \quad r \leq 2^{1/6}\sigma \quad (1)$$

and

$$U_{\text{FENE}}(r) = -0.5\kappa^b b^2 \ln(1 - r^2/b^2) \quad (2)$$

where  $\sigma$  is the diameter of a bead,  $\epsilon$  gives the strength of the LJ

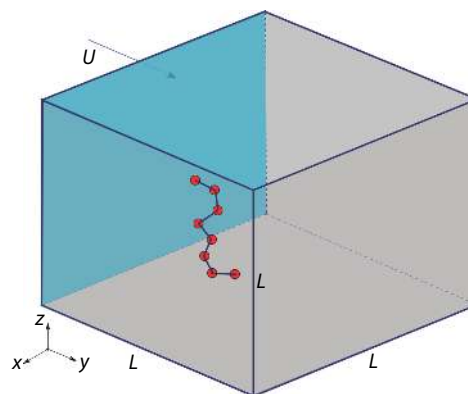


**Fig. 1** Schematic representation for the diffusion of a charged polymer in a simulation box with dimensions of  $(L, L, L)$ , where the left face of electric potential (blue) is  $U$  and the others (gray) are 0.

potential,  $\kappa^b$  represents the spring constant, and  $b$  denotes the maximum separation distance between the consecutive beads along the polymer chain.

In addition, we set the charge of beads in the polymer as  $q$  and solve the Poisson equation with the boundary electric potential to calculate the electrostatic interactions between charged beads. Therefore, we can employ the Brownian dynamics to simulate the charged polymer diffusion, which includes a noise and a friction term. Having specified the interaction potentials, the BD equation of motion for bead  $i$  is

$$\dot{\vec{r}}_i(t) = -\frac{1}{\mu} \nabla U_i + \frac{1}{\mu} \vec{f}_i(t) \quad (3)$$

where  $U_i$  is the sum of all the interaction potentials discussed above,  $\mu$  denotes the friction coefficient, and  $\vec{f}_i(t)$  is the random force related to  $\mu$  by the fluctuation dissipation theorem  $\langle \vec{f}_i(t) \vec{f}_j(t') \rangle = 2\mu k_B T \delta_{ij} \delta(t - t') \mathbf{I}$ , where  $\mathbf{I}$  is the unit tensor. We integrate the equations of charged polymer motions against a time step  $Dt$ . In our model,  $\sigma$ ,  $\epsilon$ , and  $m$  denote the units of length, energy and mass, respectively, which result in a unit of the time as  $t^* = (\epsilon/m/\sigma^2)^{1/2}$ , the charge as  $q = q^*/(4\pi\epsilon_0\sigma\epsilon)^{1/2}$ , and the electric field as  $E = E^*(4\pi\epsilon_0\sigma\epsilon)^{1/2}\sigma/\epsilon$ . We present our results in reduced units in which  $\sigma = \epsilon = m = 1$  and the temperature is  $k_B T = 1.0$ , and the friction coefficient is set as  $\mu = 0.5\tau^{-1}$ . Therefore, the dimensionless parameters of the polymer are fixed as  $\kappa^b = 30$  and  $b = 1.5$ , the charge of beads in the polymer is  $q = 10$ , and the time step is chosen as  $Dt = 0.0002$ . We fix the size of the simulation box as  $L = 80$ , the electric potential of left surface as  $U = 4$ , and the relative dielectric constant  $\epsilon_r = 80$ .

In this work, we focus on the rapid solution of electrostatic interactions between charged beads in a polymer with the external electric field, which corresponds to solving a three-dimensional Poisson equation with:

$$\nabla^2(u) = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = -\frac{\rho}{\epsilon_0\epsilon_r}, \quad (x, y, z) \in \Omega \quad (4)$$

and

$$u_{ij,k} = \alpha(x_i, y_j, z_k), \quad (x_i, y_j, z_k) \in \partial\Omega \quad (5)$$

which can be represented by the sum of a series of Dirac-Delta functions,  $u$  is the electric potential,  $\rho$  is the charge density as a function of position,  $\epsilon_0$  and  $\epsilon_r$  are the vacuum permittivity and the relative permittivity of the solvent, respectively,  $\alpha$  repres-

ents a potential function of the boundary node, and  $\Omega$  denotes a continuous domain in 3D space with boundary conditions prescribed on its boundary. As shown in Fig. 2, in order to obtain the numerical solution of  $u_{i,j,k}$ , we use the traditional 7-point central difference to discretize Eq. (4),

$$\nabla^2(u) = \frac{u_{i+1,j,k} - 2u_{i,j,k} + u_{i-1,j,k}}{(\Delta x)^2} + \frac{u_{i,j+1,k} - 2u_{i,j,k} + u_{i,j-1,k}}{(\Delta y)^2} + \frac{u_{i,j,k+1} - 2u_{i,j,k} + u_{i,j,k-1}}{(\Delta z)^2} \quad (6)$$

where  $i, j$ , and  $k$  represent the node location in the discretized mesh and  $\Delta x, \Delta y$ , and  $\Delta z$  represent the mesh spacing in the  $x, y$ , and  $z$  directions ( $\Delta x = \Delta y = \Delta z = 1$ ). The charge density of each node is estimated by a linear interpolation scheme from charged beads in the polymer. After discretization, Eq. (6) represents the linear equations, which can be expressed by a standard matrix form  $Au = B$ . Thus, we can obtain  $u_{i,j,k}$  as:

$$u_{i,j,k} = \frac{1}{6} \left( u_{i+1,j,k} + u_{i-1,j,k} + u_{i,j+1,k} + u_{i,j-1,k} + u_{i,j,k+1} + u_{i,j,k-1} + \frac{\rho_{i,j,k}}{\epsilon_0 \epsilon_r} \right) \quad (7)$$

where  $\rho_{i,j,k}$  is the charge density distribution function. In this work, we use Successive Over-Relaxation (SOR) method for solving the above equations, which is one of the most efficient iterative methods. Compared with Gauss-Seidel and Jacobi iteration methods, this method shows great advantage in calculation speed for large linear equations, whose iterative format can be written as:

$$u_{i,j,k}^{(n+1)} = u_{i,j,k}^{(n)} + \lambda \left[ \frac{1}{6} \left( u_{i+1,j,k}^{(n)} + u_{i-1,j,k}^{(n+1)} + u_{i,j+1,k}^{(n)} + u_{i,j-1,k}^{(n+1)} + u_{i,j,k+1}^{(n)} + u_{i,j,k-1}^{(n+1)} + \frac{\rho_{i,j,k}}{\epsilon_0 \epsilon_r} \right) - u_{i,j,k}^{(n)} \right] \quad (8)$$

where  $\lambda$  represents the relaxation factor and  $n$  denotes the number of iteration steps. Obviously, when  $\lambda = 1$ , the SOR is the Gauss-Seidel iteration. The optimal relaxation factor can be obtained by<sup>[30]</sup>

$$\lambda_{\text{optimal}} = 2 / [1 + \sin(\frac{\pi}{\text{Mig} - 1})] \quad (9)$$

where Mig is the minimum number of grid nodes in three directions. For  $80 \times 80 \times 80$  grids, Mig = 81,  $\lambda_{\text{optimal}} = 2 / [1 + \sin(\pi / (81 - 1))] \approx 1.924$ .

Thus, we can obtain the potential of each nodes, and further calculate the electric field in the position of charged beads by the same linear interpolation scheme, which means that we can calculate the electrostatic interactions between charged bead in the polymer with the external electric field. In order to compare the CPU time among SOR, Gauss-Seidel, and Jacobi iteration methods, we display the CPU time for these methods in Fig. 3, which includes the direct solution of Coulomb force between charged beads. The comparison results show that with the increase of charged beads, the CPU time of these methods all exhibits an increase. The SOR method is different from direct solution of Coulomb force, which is insensitive to the increase of the number of charged beads. For current grid numbers ( $80 \times 80 \times 80$ ), we find that when

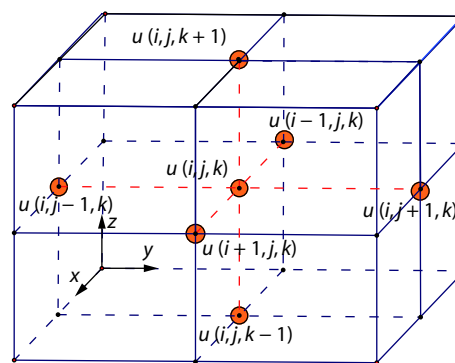


**Fig. 2** 7-Point center difference scheme:  $u_{i,j,k}$  is the potential value at the position  $(i,j,k)$ ,  $i,j,k \in (x,y,z)$ , where  $x, y, z$  are the components of the Cartesian coordinate system. The orange dash lines emphasize the strong connection between the position  $(i,j,k)$  and its neighbor points.

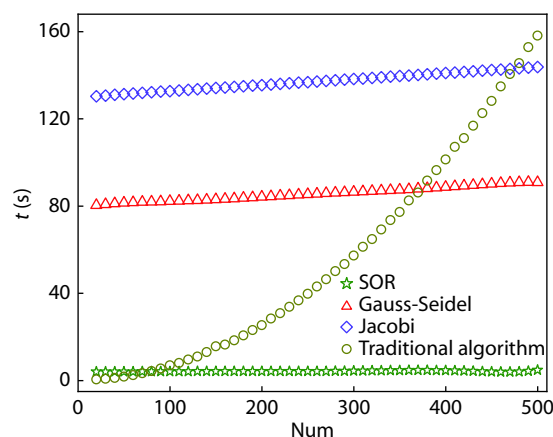


**Fig. 3** CPU time ( $t$ ) as a function of the number of charged beads in a polymer (Num) for SOR, Gauss-Seidel, and Jacobi iteration methods, and the direct solution of Coulomb force.

the number of charge beads Num is about 75, the CPU time for SOR is almost equal to that of direct solution of Coulomb force, which indicates that the SOR method is more suitable for large scale charged bead system.

In fact, in Brownian dynamics, we often need to calculate millions or even tens of millions of time steps, which means that the calculation of electrostatic interactions will take substantial time. If we only consider one charged polymer in solvents with uniform dielectric constant, within a certain time interval, the effect of the motion of charged polymer on the potential of grids far from the charged polymer can be neglected, especially in the presence of the external electric field. Thus, we can save most of the iteration time for grids far from the charged polymer, whose value of potential has almost no change within a certain time interval. This means that we can use “local iteration” instead of “global iteration” on the premise of guaranteeing calculation accuracy to save a lot of unimportant iteration time. Specially, in the simulation box, the charged beads in the polymer often occupy only a small area.

Thus, in a short simulation time, we only calculate the potential of adjacent grids and the potential of other grids is

fixed, which will be solved after several steps. For example, in Fig. 4, in order to better verify the “local iteration” method, we put the charged polymer in the middle of the simulation box. We first define a smallest “box”, which just surrounds the charged polymer. Then we expand  $L_w$  grids outwards and form a “local box”. When the “local box” exceeds the global box in a certain direction, it will not be further expanded. In addition, we set  $d_{\text{step}}$  as the interval of steps for each global iteration and a warning value ( $L_d$ ). When the nearest distance between the smallest box and the “local box” equals the warning value, we carry out the global iteration and re-divide the “local box”. Therefore, in this work, we mainly study the effects of three parameters on the calculation accuracy and speed, which include  $L_w$ ,  $d_{\text{step}}$ , and  $L_d$ . We expect that this method can greatly improve the calculation speed while ensuring the accuracy.

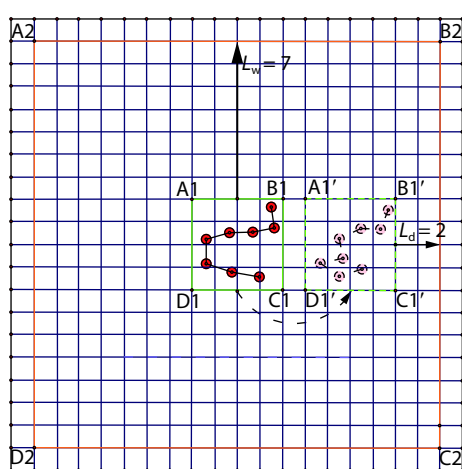


Fig. 4 2D view of the “local box”.

## RESULTS AND DISCUSSION

In the numerical experiment, all the codes use double precision arithmetic and run on a personal desktop with Intel Core i7-6700 CPU (3.40 GHz) and 16 GB RAM. We mainly study the independent effects of three parameters ( $L_w$ ,  $d_{\text{step}}$ , and  $L_d$ ) on the calculation accuracy and speed. The effect of the number of charged beads in a polymer (denoted as Num) on the calculation is also discussed. Finally, the applicability of the local iteration method is verified by the different numbers of grids. Based on the comparison between the results of local iteration and global iteration, the advantages of local iteration are explained from two aspects: on the one hand, the relative potential error ( $RE_U$ ) of each charged beads between global iteration and local iteration is calculated through:

$$RE_U = \theta_{\text{RMS}}[(U_{\text{global}} - U_{\text{local}})/U_{\text{global}}] \quad (10)$$

where  $\theta_{\text{RMS}}$  denotes the root mean square sign,  $U_{\text{global}}$  and  $U_{\text{local}}$  represent the potential at each charged beads in the global iteration and the local iteration, respectively. On the other hand, the CPU time of global iteration (denoted as G-t) and local iteration (denoted as L-t) is compared. In order to make  $L_w$ ,  $d_{\text{step}}$ ,  $L_d$ , and other parameters more universal, we use the relative size for illustration. For example,  $RL_w = L_w/L$ ,  $RS = d_{\text{step}}/\text{Step}$ ,  $RL_d = L_d/L_w$ . Because the charged beads occupy some grids, we ex-

tend  $L_w$  grids in six directions. When  $RL_w = 50\%$ , whether the “local box” completely covers the simulation box depends on the initial position of the charged beads in a polymer.

Firstly, we discuss the influence of the value of  $L_w$  in local iteration on the accuracy and CPU time, which are compared with the result of global iteration. We set the grid size as  $80 \times 80 \times 80$ , the number of charged beads in a polymer Num = 100, total step number of motion Step = 10000, the relative distance between charged polymer and boundary  $RL_d = 80\%$ , and the relative global iteration step interval RS = 0.25%, 1%, 2.5%, 5%. Under these conditions, we calculate L-t, G-t, and  $RE_U$ . As shown in Fig. 5(a), with the decrease of  $RL_w$ , the CPU time decreases, which implies an increase in the simulation efficiency. However, the decrease of  $RL_w$  has no effect on the global time G-t, because G-t only depends on the grid size and is not affected by the parameters in “local iteration” (such as  $L_w$ ,  $d_{\text{step}}$ ). Otherwise, with the same  $RL_w$ , an increasing RS also induces the decrease of CPU time. On the other hand, as shown in Fig. 5(b), both the decrease of  $RL_w$  and the increase

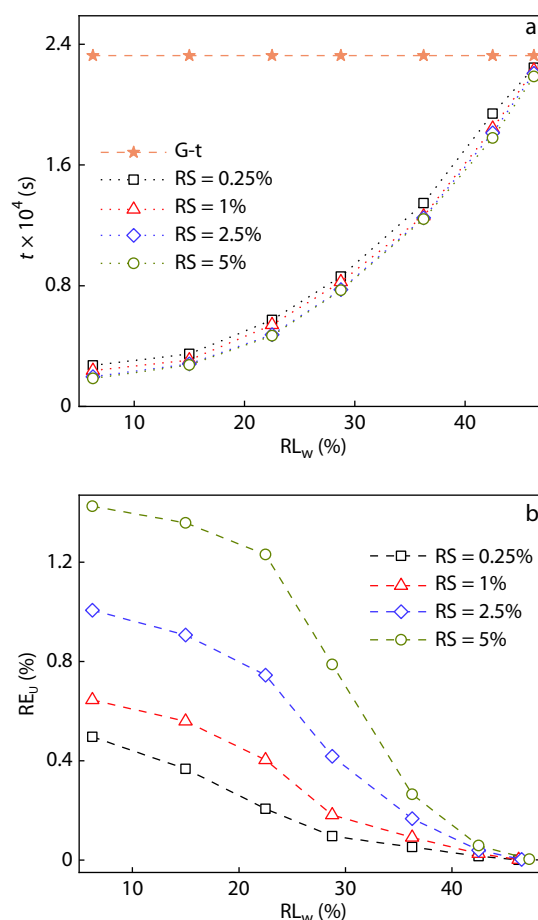


Fig. 5 (a) CPU time  $t$  and (b) the relative error of electrostatic potential  $RE_U$  as a function of the length ratio of “local box” to “global box”  $RL_w$ . G-t = 23252 s represents the global iteration time. Simulation box with dimensions of (80, 80, 80), grid spacing  $\Delta x = \Delta y = \Delta z = 1$ , grid size  $80 \times 80 \times 80$ , the number of charged beads in a polymer Num = 100, the total step number of motion Step = 10000, the relative distance between charged polymer and boundary  $RL_d = 80\%$ . RS represents the relative global iteration step.

of RS, which are of advantages to the computational efficiency, induce the increase of relative potential error  $RE_U$ . Obviously, with the increase of  $RL_w$ , the scope of local iteration is also enlarged, which implies the increase of CPU time consumed by local iteration and further results in the increase of total CPU time. At the same time, enlarged local iteration also makes the solutions approximate that of the global iteration (i.e. a smaller  $RE_U$ ). Besides, under the same total iteration step, the number of global iterations increases with RS decrease. Hence, a smaller RS indicates more computational consumption in global iteration and fewer cumulative errors, which further induces the increase of total CPU time and the reduction of relative potential error  $RE_U$ . Note that when  $RL_w$  is beyond 47%, L-t approaches G-t and  $RE_U$  approximates zero. This is because  $RL_w > 47\%$  implies that the “box” of local iteration is nearly as large as simulation box, resulting in the similar computational consumption and accuracy. Both the results of CPU time and  $RE_U$  demonstrate the availability of our “local iteration”: when  $RL_w = 6.25\%$  and  $RS = 5\%$ , the simulation efficiency is increased by 12.5 times compared to that of the traditional global iteration; meanwhile,  $RE_U$  is merely 1.43%, which is in the acceptable range. Even one wants a greater accuracy,  $RE_U \approx 0.5\%$  when  $RS = 0.25\%$ , the smaller  $RL_w$  (6.25%) also ensures a sufficient efficiency (8 times than global iteration).

In order to further observe and verify the effect of  $d_{step}$  on accuracy and speed, we use RS to explain more intuitively. Similarly, under the grid number of  $80 \times 80 \times 80$ , we set  $Num = 100$ , the length ratio of “local box” to “global box”  $RL_w = 6.25\%$ , the relative distance between charged polymer and boundary  $RL_d = 80\%$ , and Step = 10000, 15000, 20000, 30000. Then, we calculated L-t, G-t, and  $RE_U$ . As shown in Fig. 6(a), with the increase of RS, the CPU time decreases, which implies an increase in the simulation efficiency. In addition, we find that an increasing Step induces the increase of CPU time under the same RS. Otherwise, as shown in Fig. 6(b), the increase of RS, which is of advantage to the computational efficiency, would induce the increase of relative potential error  $RE_U$ . We know that with the increase of RS, the number of global iterations decreases, which implies the decrease of total CPU time. However, decreasing the value of RS can make the solution approximate that of the global iteration. We find that when  $RS = 0.25\%$ , the computational efficiency of Step = 10000 and Step = 30000 is increased by 10.1 times and 11.3 times, respectively. The result indicates that with the increase of Step, the computation efficiency of local iteration increases more obviously. In other words, the large number of steps provide more speedup than that of the small number of steps under same conditions. That  $1\% \leq RS \leq 2.5\%$  is the best under these Step is clearly exhibited in Fig. 6(b). This is due to the enough large number of global iterations even under large Step. As a result,  $RE_U$  is lower than 1.35%, and the computational efficiency can be increased by at least 9.8 times. Even one wants a greater accuracy, we can choose a larger  $RL_w$  to reduce  $RE_U$ .

Moreover, when the charged polymer moves towards the boundary of the “box” (the following “box” denotes “local box”), we also set a “warning line” —  $L_d$ . We still select the number of  $80 \times 80 \times 80$  grids, under  $Num = 100$ , Step =

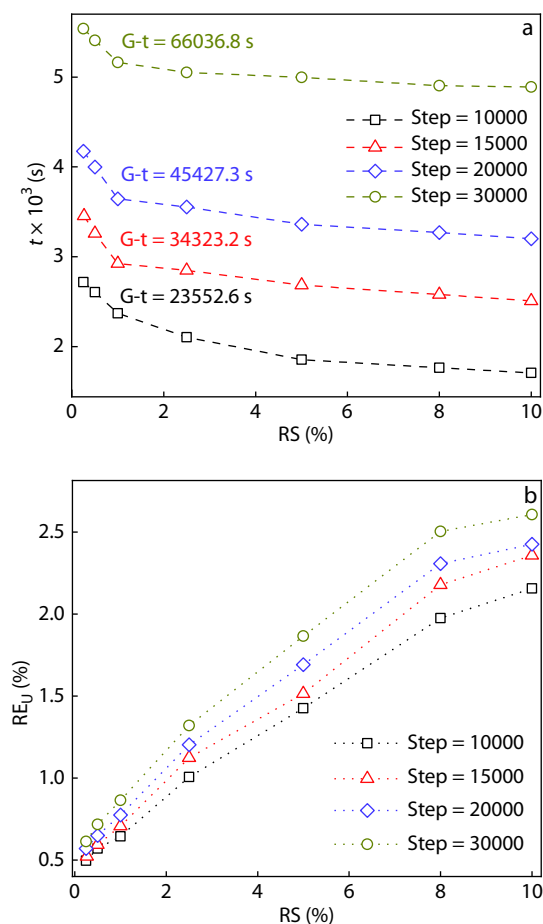


**Fig. 6** (a) CPU time  $t$  and (b) the relative error of electrostatic potential  $RE_U$  as a function of the relative global iteration step interval RS. Grid size  $80 \times 80 \times 80$ , the number of charged beads in a polymer  $Num = 100$ , the length ratio of “local box” to “global box”  $RL_w = 6.25\%$ , the relative distance between charged polymer and boundary  $RL_d = 80\%$ .

10000,  $RS = 1\%$ , and  $RL_w = 6.25\%$ , 10%, 12.5%, 15%; CPU time and  $RE_U$  corresponding to different  $RL_d$  are individually calculated. Owing to different values of  $L_w$ , the value of  $L_d$  also varies correspondingly. For example, when  $RL_w = 6.25\%$  ( $L_w = 5$ ),  $L_d$  can only take 4, 3, 2, or 1. According to Fig. 7(a), we find that the size of  $L_d$  has almost no effect on the CPU time, because  $L_d$  is only a parameter for re-dividing the “box” and it does not increase the amount of calculations like  $L_w$  and  $d_{step}$ . As shown in Fig. 7(b),  $RE_U$  decreases with the increase of  $RL_d$ . The result indicates that the accuracy increases with the increase of update frequency of “box”. This is because the effect of the charged polymer on the potential of external node of the “box” weakens with the increase of the distance between charged polymer and boundary of the “box” and  $RE_U$  thereby increases. In addition, we find that the  $RE_U/RL_d$  curve becomes steeper with the increase of  $RL_w$ . In theory,  $RE_U$  decreases with the increase of  $RL_w$ . However, we find that the accuracy of  $RL_w = 15\%$  is not as good as that of  $RL_w = 12.5\%$  when  $RL_d$  value is very small. This phenomenon is caused by the fact that the moving distance for updating the “box” increases with the increase of  $RL_w$  when  $RL_d$  is the same,

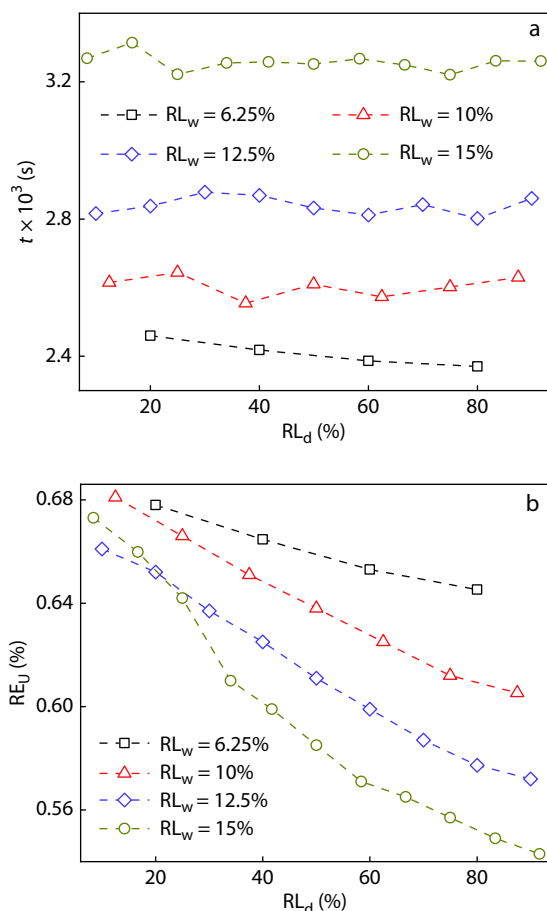


**Fig. 7** (a) CPU time  $t$  and (b) the relative error of electrostatic potential  $RE_U$  as a function of the relative distance between charged polymer and boundary  $RL_d$ . Grid size  $80 \times 80 \times 80$ , the number of charged beads in a polymer  $Num = 100$ , the total step number of motion Step = 10000, the relative global iteration step interval  $RS = 1\%$ .

leading to the lower frequency of updating the “box”, and the cumulative error further becomes too large. Such phenomenon is especially obvious when  $RL_d \leq 20\%$ .

Then, aiming at the uncertainty of the number of charged beads in polymer  $Num$  in practical problems, we choose  $Num = 100, 200, 300$ , and  $500$  under the condition that the number of grids is  $80 \times 80 \times 80$ , Step = 10000,  $RS = 1\%$ , and  $RL_d = 80\%$ . As shown in Fig. 8(a), the CPU time decreases with the decrease of  $RL_w$ , which is similar to Fig. 5(a). Moreover, we find that with the same  $RL_w$ , the increase of  $Num$  leads to the increase of CPU time for both local iteration and global iteration, while  $G-t$  of  $Num = 500$  is only 1123 s more than  $G-t$  of  $Num = 100$ . There are two reasons why the CPU time of large number of charged beads ( $L_{Num}$ ) is slightly longer than that of small number of charged beads ( $S_{Num}$ ). Firstly, the number of grids occupied by the charged polymer increases with the increasing number of charged beads in a polymer. Therefore, the local iteration range of  $L_{Num}$  is larger than that of  $S_{Num}$  even though  $RL_w$  is the same. Secondly,  $L_{Num}$  will consume more CPU time in the process of charge interpolation. However, the CPU time consumed by the interpolation calculation process is much less than the iterative calculation pro-

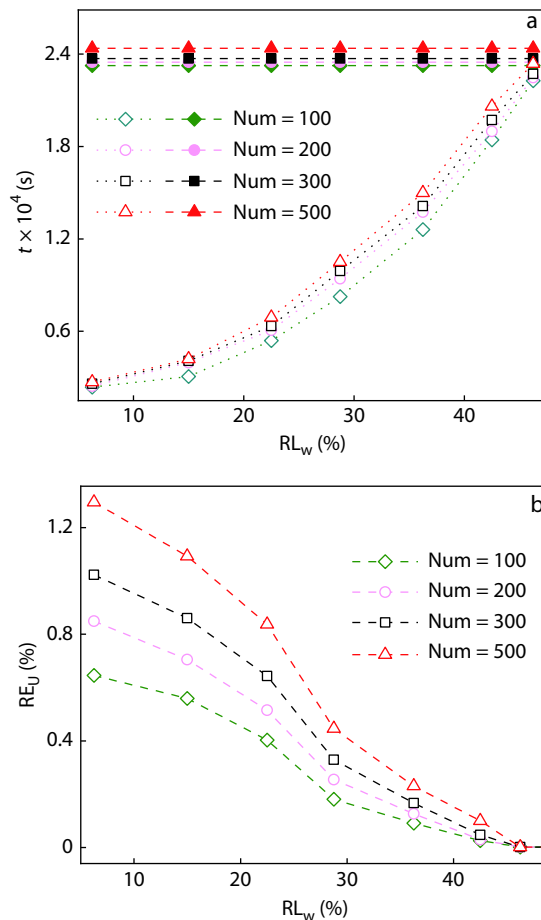


**Fig. 8** (a) CPU time  $t$  and (b) the relative error of electrostatic potential  $RE_U$  as a function of the length ratio of “local box” to “global box”  $RL_w$ . The global iteration time  $G-t = 23252, 23478, 23714$ , and  $24376$  s correspond to the cases of the charged beads number  $Num = 100, 200, 300$ , and  $500$ , respectively. Grid size  $80 \times 80 \times 80$ , the total step number of motion Step = 10000, the relative global iteration step interval  $RS = 1\%$ , the relative distance between charged polymer and boundary  $RL_d = 80\%$ .

cess. As shown in Fig. 8(b),  $RE_U$  increases slightly with the increase of  $Num$ . The reason for the difference in error is that the influence of  $L_{Num}$  on the external node potential of the “box” is greater than that of  $S_{Num}$ . We find that when  $RL_w$  is large enough, the influence of  $L_{Num}$  on the potential of external nodes is gradually weakened. We also observe that  $RE_U$  of  $L_{Num}$  and  $S_{Num}$  approach more obviously around  $RL_w = 28.75\%$ . When  $RL_w$  is between  $36\%$  and  $50\%$ ,  $RE_U$  of the four cases of  $Num$  is infinitely close. Of course, if we want to analyze a huge number of charged beads, we should expand the “local box” appropriately.

Finally, we use the “local iteration” method for testing the effect of different numbers of grids. Under the conditions of  $Num = 100$ , Step = 10000,  $RS = 1\%$ , and  $RL_d = 80\%$ , we test the  $G-t$  and  $L-t$  of  $100 \times 100 \times 100, 80 \times 80 \times 80, 100 \times 50 \times 50$ , and  $50 \times 50 \times 50$  grids and find that the values of  $G-t$  are  $46514.8, 23252.4, 6158.3$ , and  $2373.9$  s, respectively. As shown in Fig. 9(a), the CPU time decreases with the decrease of  $RL_w$ , which is similar to Fig. 5(a). Otherwise, we find that with the

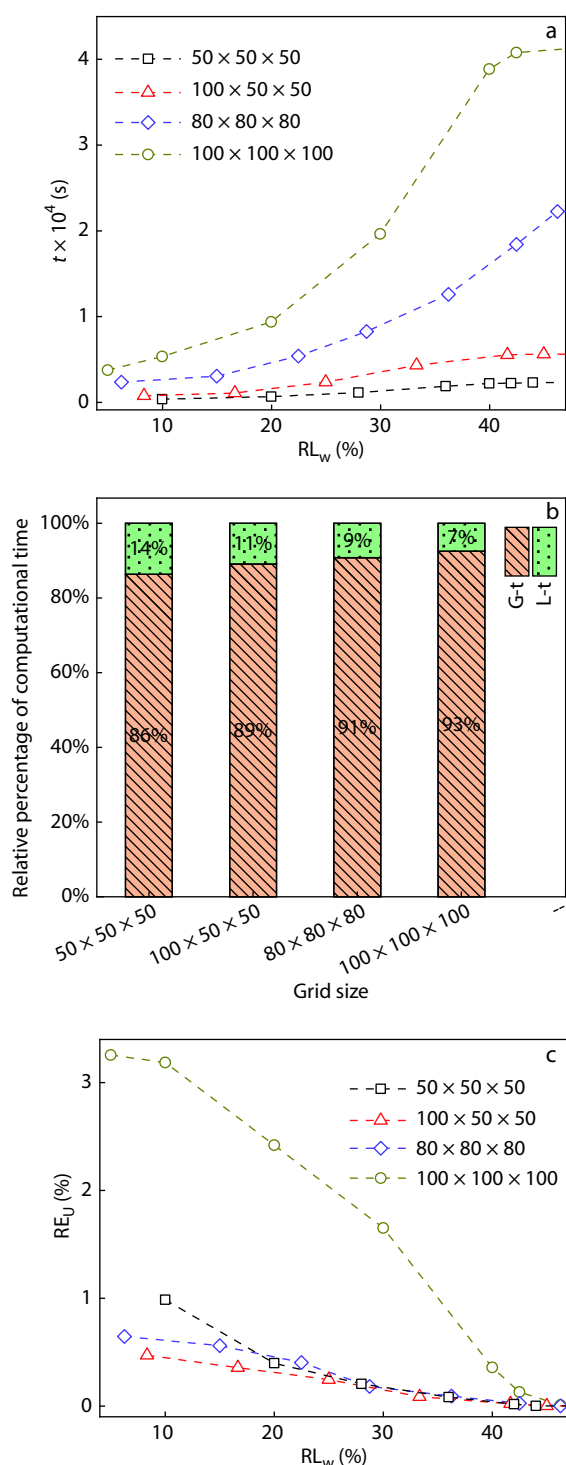


**Fig. 9** (a) CPU time  $t$  as a function of the length ratio of “local box” to “global box”  $RL_w$ . (b) The relative percentage of local iteration time L-t and global iteration time G-t when  $L_w = 5$ . (c) The relative error of electrostatic potential  $RE_U$  as a function of the length ratio of “local box” to “global box”  $RL_w$ . Simulation box with dimensions of (100, 100, 100), (80, 80, 80), (100, 50, 50), (50, 50, 50), grid spacing  $\Delta x = \Delta y = \Delta z = 1$ . The number of charged beads in a polymer Num = 100, the total step number of motion Step = 10000, the relative global iteration step interval RS = 1%, the relative distance between charged polymer and boundary  $RL_d = 80\%$ .

same  $RL_w$ , the increasing number of grids leads to the increase of CPU time. As shown in Fig. 9(b), with  $L_w = 5$ , the ratio of L-t/G-t decreases with the increase of grid number, which indicates that the efficiency of acceleration increases with the increase of grid number. The grid of  $100 \times 100 \times 100$  increases computational efficiency by 12 times more than the other cases. Our conclusion is that the “local iteration” is more suitable to computationally intensive problems. As shown in Fig. 9(c), the decrease of  $RL_w$ , which is of advantage to the computational efficiency, induces the increase of  $RE_U$ . Distinctly, the number of  $100 \times 100 \times 100$  grids is much larger than that of other sizes, and the external node increases as the “box” becomes smaller. Therefore, cumulative error caused by external nodes of the “box” leads to the larger  $RE_U$ . However, we can still control  $RE_U$  of the huge number of grids such as  $100 \times 100 \times 100$  below 3.5% with RS = 1%. In addition, we also find that  $RE_U$  of  $50 \times 50 \times 50$  is even greater than that of  $80 \times 80 \times 80$  when  $RL_w \leq 18\%$ . For smaller simulation boxes, the effect of charge on nodes outside the local iteration is greater, so  $RL_w > 18\%$  is a better choice for the small number of grids such as  $50 \times 50 \times 50$  from the Fig. 9(c).

## CONCLUSIONS

In this work, basing on the SOR iteration algorithm in the finite difference method, we present a “local iteration” method to accelerate the simulation of three-dimensional diffusion of charged polymers in solution. The effects of three parameters ( $L_w$ ,  $d_{step}$ , and  $L_d$ ) in the “local iteration” method on the computational efficiency and accuracy are further tested. Our simulation results demonstrate that both the decrease of  $RL_w$  and RS could dramatically increase the computational efficiency, while the corresponding price is the decrease of accuracy. Otherwise, the increase of  $RL_d$  could enhance the computational accuracy and has no influence on the calculative expense. Therefore, in order to reduce the effect of charged beads on the potential of external nodes of the “box”, we should make  $RL_d$  as large as possible. Accordingly, for a simulation system with Num = 100 and grid size  $80 \times 80 \times 80$ , the “local iteration” method could present an approximate 12-fold calculating speed in comparison with that of the global iteration method, where the relative potential error ( $RE_U$ ) is controlled below 1.5%.

Moreover, the increased complexity of the system (i.e., the number of charged beads) has little influence on the acceleration effect: when the same accuracy is ensured, the “local iteration” method could also reach 8-fold calculating speed for a system with Num = 500 and grid size  $80 \times 80 \times 80$ . However, the enlargement of simulation system will increase the relative potential error. In order to ensure the calculating accuracy, a larger  $RL_w$  should be used ( $RL_w \geq 25\%$ ), while the acceleration effect will decrease to only 3-fold (the relative error is controlled below 2%). Hence, the “local iteration” method needs the optimization for special problems, which includes the grid number and three parameters. This could significantly accelerate the simulation processes and guarantee precision control, thereby efficiently accelerating the simulation processes. In practical problems, we may also encounter inhomogeneous media. Therefore, it is very meaningful to further optimize different types of three-dimensional Poisson equation algorithm.

## ACKNOWLEDGMENTS

This work was financially supported by the National Natural Science Foundation of China (No. 21604086), Jilin Provincial science and technology development program (No. 20190103124JH), and Key Research Program of Frontier Sciences, CAS (No. QYZDY-SSW-SLH027).

## REFERENCES

- Ermak, D. L.; McCammon, J. A. Brownian dynamics with hydrodynamic interactions. *J. Chem. Phys.* **1978**, *69*, 1352–1360.
- Northrup, S. H.; Alison, S. A.; McCammon, J. A. Brownian dynamics simulation of diffusion-influenced bimolecular reactions. *J. Chem. Phys.* **1984**, *80*, 1517–1526.
- Zhou, H. X. On the calculation of diffusive reaction-rates using brownian dynamics simulations. *J. Chem. Phys.* **1990**, *92*, 3092–3095.
- Luty, B. A.; McCammon, J. A.; Zhou, H. X. Diffusive reaction-rates from brownian dynamics simulations-replacing the outflow surface by an analytical treatment. *J. Chem. Phys.* **1992**, *97*, 5682–5686.
- Bauler, P.; Huber, G.; Leyh, T.; McCammon, J. A. Channeling by proximity: the catalytic advantages of active site colocalization using brownian dynamics. *J. Phys. Chem. Lett.* **2010**, *1*, 1332–1335.
- McGuffee, S. R.; Elcock, A. H. Diffusion, crowding & protein stability in a dynamic molecular model of the bacterial cytoplasm. *PLoS Comput. Biol.* **2010**, *6*, 1000694.
- Toukmaji, A. Y.; Board, J. A. Jr. Ewald summation techniques in perspective: A survey. *Comput. Phys. Commun.* **1996**, *95*, 73–92.
- Deserno, M.; Holm, C. How to mesh up Ewald sums. II. An accurate error estimate for the particle-particle-mesh algorithm. *J. Chem. Phys.* **1998**, *109*, 7694–7701.
- Zhou, S.; Sun, H.; Cheng, L. T.; Dzubiella, J.; Li, B.; McCammon, J. A. Stochastic level-set variational implicit-solvent approach to solute-solvent interfacial fluctuations. *J. Chem. Phys.* **2016**, *145*, 054114.
- Darden, T.; York, D.; Pedersen, L. Particle mesh Ewald: an  $N \log(N)$  method for Ewald sums in large systems. *J. Chem. Phys.* **1993**, *98*, 10089–10092.
- Ballenegger, V.; Cerda, J. J.; Holm, C. How to convert SPME to P3M: influence functions and error estimates. *J. Chem. Theory Comput.* **2012**, *8*, 936–947.
- Antia, H. M. *Numerical methods for scientists and engineers*. Vol. 2. Springer, **2012**.
- Huber, G. A.; Miao, Y.; Zhou, S.; Li, B.; McCammon, J. A. Hybrid finite element and Brownian dynamics method for charged particles. *J. Chem. Phys.* **2016**, *144*, 164107.
- Smith, G. D. *Numerical solution of partial differential equations*. Oxford University Press, **1985**.
- Kamboh, S. A.; Labadin, J.; Rigit, A. R. H. Computational modeling and simulation of EHD ion-drag pumping using finite difference method. *Simul. Mater. Sci. Eng.* **2013**, 207–211.
- Spotz, W. F.; Carey, G. F. A high-order compact formulation for the 3D Poisson equation. *Numer. Meth. Part. D. E.* **2015**, *12*, 235–243.
- Gupta, M.; Kouatchou, J. Symbolic derivation of finite difference approximations for the three-dimensional Poisson equation. *Numer. Meth. Part. D. E.* **1998**, *14*, 593–606.
- Wang, J.; Zhong, W.; Zhang, J. A general mesh size fourth-order compact difference discretization scheme for 3D Poisson equation. *Appl. Math. Comput.* **2006**, *183*, 804–812.
- Wang, H.; Zhang, Y.; Ma, X.; Qiu, J.; Liang, Y. An efficient implementation of fourth-order compact finite difference scheme for Poisson equation with Dirichlet boundary conditions. *Appl. Math. Comput.* **2016**, *71*, 1843–1860.
- Tritsiklis, J. N. A comparison of Jacobi and Gauss-Seidel parallel iterations. *Appl. Math. Lett.* **1998**, *2*, 167–170.
- Kamboh, S. A.; Labadin, J.; Rigit, A. R. H. Computational time analysis of the numerical solution of 3D electrostatic Poisson's equation. *International Conference on Mathematics, Engineering and Industrial Applications (ICoMEIA)*. **2015**.
- Fukuchi, T. Finite difference method and algebraic polynomial interpolation for numerically solving Poisson's equation over arbitrary domains. *AIP Adv.* **2014**, *4*, 060701.
- Jozwikowska, A. Numerical solution of the nonlinear Poisson equation for semiconductor devices by application of a diffusion-equation finite difference scheme. *J. Appl. Phys.* **2008**, *104*, 63715.
- Yang, X. I. A.; Mittal, R. Acceleration of the Jacobi iterative method by factors exceeding 100 using scheduled relaxation. *J. Comput. Phys.* **2014**, *274*, 695–708.
- Adsuara, J. E.; Cordero-Carrion, I.; Cerda-Duran, P.; Aloy, M. A. Scheduled relaxation Jacobi method: improvements and applications. *J. Comput. Phys.* **2016**, *321*, 369–413.
- Richardson, L. F. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philos. R. Soc. London, Ser.* **1911**, *201*, 307–357.
- Young, D. Iterative methods for solving partial difference equations of elliptic type. *T. Am. Math. Soc.* **1954**, *76*, 92–111.
- Schultz, M. *Analysis of numerical methods*. Dover Publications, **1994**.
- Juncosa, M. L.; Mulikin, T. W. On the increase of convergence rates of relaxation procedures for elliptic partial difference equations. *JACM* **1960**, *7*, 29–36.
- Adsuara, J. E.; Cordero-Carrion, I.; Cerda-Duran, P.; Mewes, V.; Aloy, M. A. On the equivalence between the Scheduled Relaxation Jacobi method and Richardson's non-stationary method. *J. Comput. Phys.* **2017**, *332*, 446–460.
- Pratapa, P. P.; Suryanarayana, P.; Pask, J. E. Anderson acceleration of the Jacobi iterative method: an efficient alternative to Krylov methods for large, sparse linear systems. *J. Comput. Phys.* **2016**, *306*, 43–54.
- Ge, Y. Multigrid method and fourth-order compact difference discretization scheme with unequal meshsizes for 3D Poisson equation. *J. Comput. Phys.* **2010**, *229*, 6381–6391.
- Dai, R.; Lin, P.; Zhang, J. An EXCMG accelerated multiscale multigrid computation for 3D Poisson equation. *Comput. Math. Appl.* **2019**, *77*, 2051–2060.
- Sutmann, G.; Steffen, B. High-order compact solvers for the three-dimensional Poisson equation. *J. Comput. Appl. Math.* **2006**, *187*, 142–170.
- Kouatchou, J.; Zhang, J. Optimal injection operator and high order schemes for multigrid solution of 3D Poisson equation. *Int. J. Comput. Math.* **2000**, *76*, 173–190.
- Zhang, J. Fast and high accuracy multigrid solution of the three dimensional Poisson equation. *J. Comput. Phys.* **1998**, *143*, 449–461.
- Frenkel, D.; Smit, B. *Understanding molecular simulation*. Academic Press, New York, **2002**.
- Allen, M.; Tildesley, D. *Computer simulation of molecular liquids*. Clarendon Press, Oxford, **1981**.
- Maitland, G. C. *Intermolecular forces: their origin and determination*. Oxford University Press, Oxford, **1981**.
- Warner, Harold R. Kinetic theory and rheology of dilute suspensions of finitely extendible dumbbells. *Industrial and Engineering Chemistry, Fundamentals* **1972**, *11*, 379–387.