Available online at www.prace-ri.eu

Partnership for Advanced Computing in Europe

Development of a Grid-based Electrostatics Model in a Memory-Distributed Manner for DL_POLY_4

P. Petkov[a]*, I. Todorov[b], D. Grancharov[a], N. Ilieva[a], E. Lilkova[a], L. Litov[a], S. Markov[a]

[a]*NCSA, Akad. G. Bonchev 25A, Sofia 1311, Bulgaria*
[b]*STFC Daresbury Laboratory, Daresbury, Warrington WA4 4AD, UK*

Abstract

Electrostatic interactions in molecular simulations are usually evaluated by employing the Ewald summation method, which splits the summation into a short-ranged part, treated in real space and a long-range part treated in reciprocal space. For performance purposes in molecular dynamics software the latter is usually handled by SPME or P^3M grid based methods both relying on 3D fast Fourier transform (FFT) as their central operation. However, the Ewald summation method is derived for model systems that are subject to 3D periodic boundary conditions (PBC) while there are many models of scientific as well as commercial interest, where geometry implies a 1D or 2D structures. Thus for systems, such as membranes, interfaces, linear protein complexes, thin layers, nanotubes, etc.; employing Ewald summation based techniques is either very disadvantageous computationally or impossible at all. Another approach to evaluate the electrostatics interactions is to solve the Poisson equation of the model-system charge distribution on a 3D special grid. The formulation of the method allows an elegant way to switch on and off the dependence on periodic boundary conditions in a simple manner. Furthermore, 3D FFT kernels are known to scale poorly at large scale due to excessive memory and communication overheads, which makes the Poisson solvers a viable alternative for DL_POLY on the road to exascale. This paper describes the work undertaken to integrate a Poisson solver library, developed in PRACE-2IP [1], within the DL_POLY_4 domain decomposition framework. The library relies on a unique combination of bi-conjugated gradient (BiCG) and conjugated gradient (CG) methods to warrant both independence on initial conditions with a rapid convergence of the solution on the one hand and stabilization of possible fluctuations of the iterative solution on the other. The implementation involves the development of procedures for generating charge density and electrostatic potential grids in real space over all domains in a distributed manner as well as halo exchange routines and functions to calculate the gradient of the potential in order to recover electrostatic forces on point charges.

Introduction

Due to the long-range nature of the electrostatic interactions, their treatment is of crucial importance in the context of studying the materials' properties at atomistic and molecular level by means of molecular dynamics (MD) simulations. The electrostatics acts on polar or charged molecules, including water, ions, amino acids, nucleic acids, carbohydrates, and lipids, and hence greatly influences their behaviour and properties, including solvation, folding and binding. Thus, it is essential that the electrostatic interactions be evaluated with high level of accuracy in MD simulations. However, their long-range and many-body character makes their evaluation immensely computationally demanding – in fullness, this is a N^2 problem, N being the number of particles in the model system. A number of techniques have been developed to tract the problem in a better scalable manner, of which the most

*Corresponding author. *E-mail address*: *peicho.petkov@gmail.com* widely used, especially in computational bio-chemistry and bio-physics, are SPME [2,3] and P^{3}M [4]. Both are based on the Ewald summation method and use Fourier transforms as a central operation. The application of these Ewald based techniques is however limited to model systems with 3D periodic boundary conditions (PBC) in its original formulation. An alternative approach is to use a continuum electrostatics model that allows simulations under general boundary conditions, for example by solving the Poisson's equation. Such a technique can be efficiently employed to evaluate accurately the electrostatic contribution to the free energy [5], for modeling of biomolecular titration states [6], computation of the electrostatic potential of lipid bilayers [7], ion-channel characteristics calculations [8], etc.

In this paper, we present some of the most important aspects of the work undertaken on DL_POLY_4.05 to implement a parallel Poisson's Equation Solver for model systems without PBC using a 27-stencil discretization scheme based on the stabilised bi-conjugate gradient (BiCG) method. The work has led to a sub-program module, which includes procedures for generating charge density and electrostatic potential grids in real space in a commensurate manner with the domain decomposition strategy adopted in DL_POLY_4. The module also includes subroutines for halo exchange information and functions to calculate the gradient of the potential in order to recover electrostatic forces on charged atoms. It is worth noting that further work was undertaken to properly account for the electrostatics in model systems with intra-molecular interaction since while the Poisson solver returns a global electrostatic potential locally a number of direct interactions may be suppressed due to intra-molecular bonding.

The motivation for this work was to enhance DL_POLY_4 capability in terms of (i) enabling it to tract accurately and efficiently electrostatic interaction under non-periodic boundary conditions, and (ii) preparation for the exascale involving simulations of very large (~10^7 atoms) systems on large core counts (~10^5), where the FFT component of SPME and P^{3}M is known to lose its scalable performance due to excessive non-local communications and large memory requirements.

Theory

The Poisson equation gives the electrostatic potential $\phi(\vec{r})$ created by a charge distribution $\rho(\vec{r})$ in a continuum medium with a given dielectric constant $\epsilon(\vec{r})$:

$$\nabla^2\phi = -\frac{\rho}{\epsilon} \tag{1}$$

Let us denote (i,j,k) three-dimensional lattice indices and $(\phi_{\alpha\alpha})_{ijk}$ is an approximation to the second partial derivative with respect to the coordinate direction $\alpha\alpha$. Then the discretized form of Eq. (1) can be written as:

$$(\phi_{xx})_{ijk} + \left(\phi_{yy}\right)_{ijk} + (\phi_{zz})_{ijk} = -f_{ijk}. \tag{2}$$

We denote the second-order difference operator as:

$$(\phi_{\alpha\alpha})_{ijk} = \frac{1}{{h_\alpha}^2}\delta_\alpha^2\phi_{ijk}, \tag{3}$$

where h_α is the grid spacing in direction α and δ_α^2 is the second-order difference operator

$$\begin{aligned}\delta_x^2 &= \phi_{i-1,j,k} - 2\phi_{i,j,k} + \phi_{i+1,j,k},\\ \delta_y^2 &= \phi_{i,j-1,k} - 2\phi_{i,j,k} + \phi_{i,j+1,k},\\ \delta_z^2 &= \phi_{i,j,k-1} - 2\phi_{i,j,k} + \phi_{i,j,k+1}.\end{aligned}$$

We are using a 27-point stencil finite difference approximation. Given this approximation and $\mathrm{h} = 0.5\text{Å}$; $\mathrm{i} = 1,2,\dots\mathrm{m}$; $\mathrm{j} = 1,2,\dots\mathrm{n}$; $\mathrm{k} = 1,2,\dots\mathrm{p}$; $\mathrm{m} = \mathrm{x/h}$; $\mathrm{n} = \mathrm{y/h}$; $\mathrm{p} = \mathrm{z/h}$; $\rho = \mathrm{q/h^3}$, where q is the point electric charge, we arrive at the following equation for the unknowns $\phi(i,j,k)$ using when the finite difference operator from Eq.(3):

$$\begin{aligned}144h^2\rho = {} & -600\varphi_{i,j,k} + 60\left[\varphi_{i,j,k-1} + \varphi_{i,j,k+1}\right] + 60\left[\varphi_{i-1,j,k} + \varphi_{i+1,j,k} + \varphi_{i,j-1,k} + \varphi_{i,j+1,k}\right] + \\ & 18\left[\varphi_{i-1,j-1,k} + \varphi_{i-1,j+1,k} + \varphi_{i+1,j-1,k} + \varphi_{i+1,j+1,k}\right] + 18\left[\varphi_{i-1,j,k-1} + \varphi_{i+1,j,k-1} + \varphi_{i-1,j,k+1} + \right. \\ & \left. \varphi_{i+1,j,k+1} + \varphi_{i,j-1,k-1} + \varphi_{i,j+1,k-1} + \varphi_{i,j-1,k+1} + \varphi_{i,j+1,k+1}\right] + 3\left[\varphi_{i+1,j-1,k-1} + \varphi_{i-1,j-1,k-1} + \right. \\ & \left. \varphi_{i+1,j+1,k-1} + \varphi_{i+1,j+1,k+1} + \varphi_{i-1,j-1,k+1} + \varphi_{i+1,j-1,k+1} + \varphi_{i-1,j+1,k-1} + \varphi_{i-1,j+1,k+1}\right]\end{aligned} \tag{4}$$

The maximum absolute error of this approximation for $\varphi(\vec{r})$ is $O(h^{-6})$.

Eq. (4) can be rewritten in the following matrix form:

$$Au(s) = b(s), \tag{5}$$

where

$$A = \begin{pmatrix} R & S & & \cdots & & & \\ S & R & S & & & & \\ & \vdots & & \ddots & & \vdots & \\ & & & \cdots & S & R & S \\ & & & & & S & R \end{pmatrix}$$

$$R = \begin{pmatrix} R_1 & R_2 & & \cdots & & & \\ R_2 & R_1 & R_2 & & & & \\ & \vdots & & \ddots & & \vdots & \\ & & & \cdots & R_2 & R_1 & R_2 \\ & & & & & R_2 & R_1 \end{pmatrix}, \qquad S = \begin{pmatrix} S_1 & S_2 & & \cdots & & & \\ S & S_1 & S_2 & & & & \\ & \vdots & & \ddots & & \vdots & \\ & & & \cdots & S & S_1 & S_2 \\ & & & & & S_2 & S_1 \end{pmatrix}$$

Matrix *A* has *p* blocks and each block is of order $mn \times mn$. *R* and *S* have *n* blocks and each block is of order $m \times m$.

$$R_1 = \begin{pmatrix} -600 & 60 & & \cdots & & & \\ 60 & -600 & 60 & & & & \\ & \vdots & & \ddots & & \vdots & \\ & & & \cdots & 60 & -600 & 60 \\ & & & & & 60 & -600 \end{pmatrix}, \qquad R_1 = \begin{pmatrix} 60 & 18 & & \cdots & & & \\ 18 & 60 & 18 & & & & \\ & \vdots & & \ddots & & \vdots & \\ & & & \cdots & 18 & 60 & 18 \\ & & & & & 18 & 60 \end{pmatrix}$$

$$S_1 = \begin{pmatrix} 60 & 18 & & \cdots & & & \\ 18 & 60 & 18 & & & & \\ & \vdots & & \ddots & & \vdots & \\ & & & \cdots & 18 & 60 & 18 \\ & & & & & 18 & 60 \end{pmatrix}, \qquad S_1 = \begin{pmatrix} 18 & 3 & & \cdots & & & \\ 3 & 18 & 3 & & & & \\ & \vdots & & \ddots & & \vdots & \\ & & & \cdots & 3 & 18 & 3 \\ & & & & & 3 & 18 \end{pmatrix}$$

$$u = \begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_p \end{pmatrix} \qquad b = \begin{pmatrix} B_1 \\ B_2 \\ B_3 \\ \vdots \\ B_p \end{pmatrix}$$

$$U_k = (\, u_{1,1,k}\,, u_{2,1,k}\,, \dots, u_{m,1,k}\,;\ u_{1,2,k}\,, u_{2,2,k} \,..\, u_{m,2,k}\ \cdots\ u_{1,n,k}\,, u_{2,n,k}\,, u_{m,n,k}\,)$$

$$B_k = (\, b_{1,1,k}, b_{2,1,k}\,, \dots, b_{m,1,k}\,;\ b_{1,2,k}, b_{2,2,k}\,, \dots, b_{m,2,k} \cdots b_{1,n,k}\,, b_{2,n,k}\,, b_{m,n,k}\,);$$

If in the point with lattice indices t, s, v the charge is q, then the charge density is $b_{t,s,v} = \rho = q/h^3$.
We use an implementation of a parallel BiCGSTAB algorithm [9] to find the solution of the aforementioned system of linear algebraic equations. The multiplications of the matrix *A* with the vectors of the solution φ and the search directions s are the most time consuming part of the algorithm. The number of arithmetic operations "multiplication and addition" is $27mnp$ for each of the two multiplications.

Results

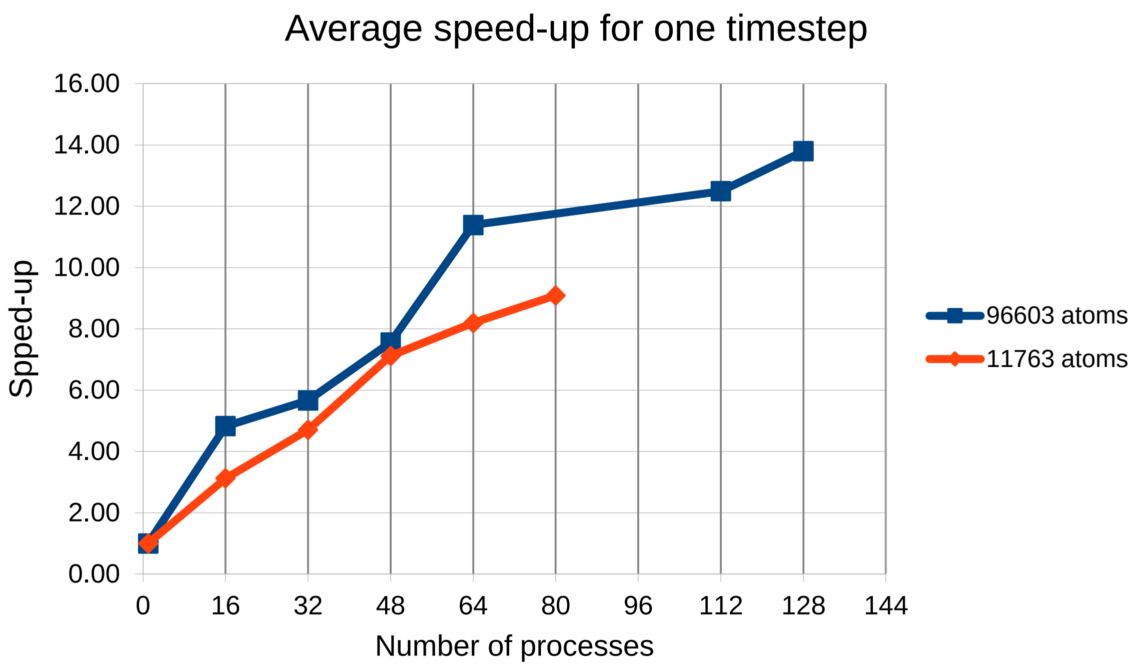
We tested the performance and scalability of the DL_POLY Poisson solver module on two model systems with 3921 (~100 Å box edge) and 32201 (~200 Å box edge) TIP3P water molecules, respectively. The models systems intermolecular interactions (van der Waals and Coulomb) were handled with a smaller cutoff (of 5 Å instead of 8-10 Å) than usual in order to warrant a linear scaling of the linked-cells algorithm for construction the Verlet neighbour list (VNL) and have the Poisson solver routine as the dominant contribution of the force evaluation cycle. This assumption will hold true up to 64 and 1024 MPI counts for the small and the large system respectively. The simulations were performed on a Linux cluster with Intel® Xeon® E5540 @ 2.53GHz chips. The reported results were obtained using the *GNU* Fortran90 compiler, *gfortran* version 4.8.1, with O3 level of optimisation and the execution times were averaged over 20 timesteps.

Table 1 lists the performance data for the models systems run under the conditions outlined above. The missing values indicate run failures due to grid size mismatch on the selected processor counts as processor grids dictated unfavourable grid cell sizes for the linear 27-stencil approximation of the potential gradients. The data are plotted in Figure 1, which indicates that the developed method has a close to linear performance at small processor counts with some degradation as expected at large ones. As the algorithms work in good scaling regime the performance degradation is clearly due to the increase of the volume of messages of smaller sizes as the processor counts increase.

Table 1. Execution time per timestep

Number of processes	**11763 atoms**		**96603 atoms**	
	Time, [s]	**Speed-up**	**Time, [s]**	**Speed-up**
1	0.368	1.00	0.632	1.00
16	0.118	3.12	0.131	4.82
32	0.078	4.70	0.112	5.67
48	0.052	7.12	0.084	7.56
64	0.045	8.19	0.056	11.39
80	0.041	9.09	N/A	N/A
112			0.051	12.49
128			0.046	13.80

Figure 1. Execution time as a function of the number of the MPI processes for one timestep.



It is worth noting that for exascale performance we expect system sizes of the order of 10^8 particles on 10^5 core counts. This relates the performance of the Poisson solver as running the large test system on 96 cores or the small one on 12 cores. Under these conditions, we indeed stay in linear scaling regimes. Furthermore, the Poisson solver routines already contain openMP parallelism and when DL_POLY_4 allows for the use of openMP within its MPI framework of domain decomposition the scalability will be further improved by offsetting the MPI degradation by loading more work within a domain carried out by the domain's openMP threads.

Conclusions

The motivation for this work was to enhance DL_POLY_4 capability in terms of (i) enabling it to treat accurately and efficiently electrostatic interaction under non-periodic boundary conditions, and (ii) to prepare for exascale computing, which will involve simulations of very large (~10^7 atoms) systems on machines with a ~billion-way concurrency, where the FFT component of SPME and P^3M methods is already known to lose its scalable efficiency very quickly and exhibit poor performance due to excessive non-local communications and large memory requirements. In this paper we demonstrated that the real-space Poisson solver methodology we developed provides a viable alternative to the aforementioned methods which is especially indispensible in the case of model systems with no periodic boundary conditions. The Poisson solver worked in truly memory distributed manner with very good hard scaling performance for the systems sizes investigated. The decline in performance at large processor counts was due to limits of hard scaling where communication becomes dominant over computation due to increase of small size messages volume and decrease of compute per core.

We believe that this particular Poisson solver methodology is very promising and could be improved further by adopting the use of cardinal B-splines. This will lead to smoother initial conditions of the charge density and thus faster and more stable electrostatic potential solutions as well as more accurate energy, stress and force evaluations. This will potentially decrease compute prefactors and improve the method's scalable performance – an option worth pursuing elsewhere.

References

[1] Markov S., Petkov P., Grancharov D. and Georgiev G., High Performance Poisson Equation Solver for Hybrid CPU/GPU Systems, PRACE Whitepaper No 112, http://www.prace-ri.eu/IMG/pdf/wp112.pdf

[2] Darden T., York D., Pedersen L., Particle mesh Ewald: An N log(N) method for Ewald sums in large systems. The Journal of Chemical Physics. 98(12):10089–10092 (1993).

[3] Bush I. J., Todorov I. T., Smith W., A DAFT DL_POLY distributed memory adaptation of the Smoothed Particle Mesh Ewald method. Comp. Phys. Commun. 175, 323–329 (2006).

[4] Roger W. Hockney, James W. Eastwood, Particle-Particle-Particle-Mesh (P3M) Algorithms. Computer simulation using particles. (CRC Press, 1988, pp. 267–304).

[5] Prabhu N., Sharp K., Protein-Solvent Interactions. Chemical Reviews, 106(5):1616–1623 (2006).

[6] Bombarda E., Ullmann M.M., pH-Dependent pKa Values in Proteins – A Theoretical Analysis of Protonation Energies with Practical Consequences for Enzymatic Reactions. The Journal of Physical Chemistry B, 114(5):1994–2003 (2010).

[7] Gurtovenko A. A. and Vattulainen I., Calculation of the electrostatic potential of lipid bilayers from molecular dynamics simulations: Methodological issues, J. Chem. Phys. 130, 215107 (2009).

[8] Roux B., Allen T., Berneche S., Im W., Theoretical and computational models of biological ion channels. Quarterly Reviews of Biophysics, 37(1):15–103 (2004).

[9] Van der Vorst, H. A., Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems. SIAM J. Sci. and Stat. Comput. 13 (2): 631–644 (1992).

Acknowledgements

This work was financially supported by the PRACE-3IP project funded in part by the EUs 7th Framework Programme (FP7/2007-2013) under grant agreement no. RI-312763.