Journal of Molecular Graphics and Modelling 50 (2014) 44–49

Contents lists available at ScienceDirect

Journal of Molecular Graphics and Modelling

journal homepage: www.elsevier.com/locate/JMGM

Topical Perspectives

VCMM: A visual tool for continuum molecular modeling

Shiyang Bai, Benzhuo Lu*

LSEC, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Accepted 8 March 2014
Available online 28 March 2014

Keywords:
Continuum molecular modeling
Visualization
Visual analysis
Unstructured mesh
Numerical solvers

ABSTRACT

This paper describes the design and function of a visualization tool, VCMM, for visualizing and analyzing data, and interfacing solvers for generic continuum molecular modeling. In particular, an emphasis of the program is to treat the data set based on unstructured mesh as used in finite/boundary element simulations, which largely enhances the capabilities of current visualization tools in this area that only support structured mesh. VCMM is segmented into molecular, meshing and numerical modules. The capabilities of molecular module include molecular visualization and force field assignment. Meshing module contains mesh generation, analysis and visualization tools. Numerical module currently provides a few finite/boundary element solvers of continuum molecular modeling, and contains several common visualization tools for the numerical result such as line and plane interpolations, surface probing, volume rendering and stream rendering. Three modules can exchange data with each other and carry out a complete process of modeling. Interfaces are also designed in order to facilitate usage of other mesh generation tools and numerical solvers. We develop a technique to accelerate data retrieval and have combined many graphical techniques in visualization. VCMM is highly extensible, and users can obtain more powerful functions by introducing relevant plug-ins. VCMM can also be useful in other fields such as computational quantum chemistry, image processing, and material science.

1. Introduction

In biomolecular simulation studies, the explicit solvent methods treat the solvent in full atomic detail, and the implicit solvent methods represent the solvent through its average effect on the solute. Explicit solvent methods are demanding a large number of freedom because they offer a very detailed description of the solvent and ions, whereas implicit solvent methods enjoy the advantage of reduced degrees of freedom. The Poisson–Boltzmann (PB) equation represents a typical implicit solvent model, and provides a simplified continuum description of the discrete particle (e.g., water, ion, and even protein molecule) distributions in solution, as well as the electrostatic interaction of a solvated system at equilibrium state [1–3]. The Poisson–Nernst–Planck (PNP) model provides a continuum description of the non-equilibrium electrodiffusion process of ion transport in solution [4].

Finite difference and finite volume methods dominate the methodologies used in PB solvers in biochemical and biophysical communities. These solvers are based on structured meshes, and the molecular surface is used to define the map of the dielectric function. However, the position of the molecular surface is usually not precisely computed and constructed, and the normal direction of the molecular surface is not calculated, which leads to a neglect of the continuity conditions on the solution. In recent years, there are much research efforts on the boundary element methods (BEM) and finite element methods (FEM) (e.g., see [2,5–13]). With the presence of surface mesh conforming to the molecular boundary, the solutions of these methods automatically satisfy the continuity conditions at the molecular boundary, hence leading to more accurate results. Unlike finite difference methods using structured grid, those methods are based on unstructured mesh, which will lead to many new issues in visualization and data analysis/management.

Visual analysis is an important part of scientific computations. There have been many tools for molecular visualization. Pymol [14] and VMD [15] are among the most popular ones nowadays. GRASP is another program with particular emphasis on the display and manipulation of the surfaces of molecules and their electrostatic properties [16]. However, these software packages lack the capability of unstructured mesh management and visual analysis of numerical results based on unstructured mesh. Furthermore, these packages do not provide functions for BEM and FEM simulations. Therefore, a molecular visualization tool with these functions is an urgent need.

* Corresponding author.Tel.: +86 1082541904.
E-mail addresses: baishiyang@lsec.cc.ac.cn (S. Bai), bzlu@lsec.cc.ac.cn (B. Lu).

http://dx.doi.org/10.1016/j.jmgm.2014.03.006
1093-3263/

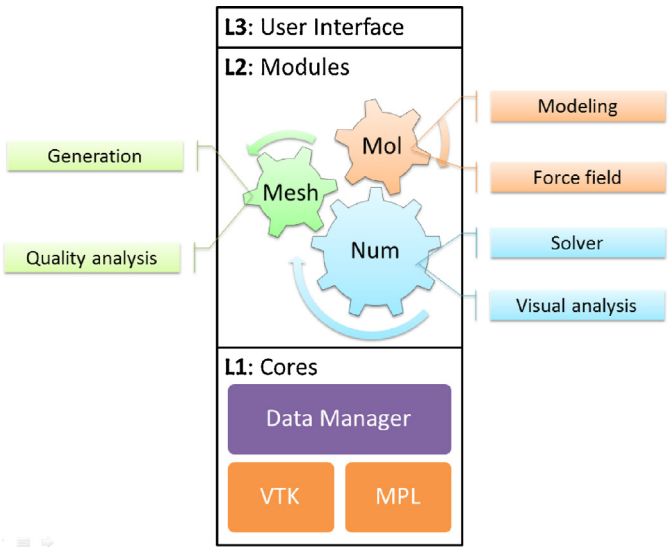

Fig. 1. Schematic design of the VCMM architecture.

This paper will report a general visual tool, VCMM, for continuum molecular modeling with focal applications of BEM and FEM. We developed a mesh dividing technique based on molecular structure to manage the large molecular mesh, which enables VCMM to have a high efficiency of handling molecular mesh. A variety of visual analysis tools are designed for the numerical results of BEM and FEM solvers. To facilitate the simulation processes, VCMM has integrated into a tool chain. The tool chain includes the following parts:

1 Molecular visualization and force field tools.
2 Surface and volume mesh generation tools.
3 Mesh visualization and analysis tools.
4 BEM and FEM solvers.
5 Numerical results visualization and analysis tools.

The organization of this article is as follows. The design and overview of VCMM are introduced first, followed by a description of the features of each module. We then also give a demonstrating example including a whole process of molecular electrostatic modeling. Finally, we discuss the future outlook, list the availability, and end up with acknowledgments.

2. Design and overview

The frame of VCMM is written in Python and the cores are written in C++. VCMM is designed as a layered architecture as showned in Fig. 1. The first layer contains a data manager and two visual packages: visualization toolkit (VTK) [17] and Matplotlib (MPL) [18]. The data manager is designed to handle various types of data (e.g., PDB [19], PQR, OFF, MESH, and VTK.) and is used in each module and visualization. The second layer is a logical layer, which can control the simulation process. This layer consists of three modules: molecular module, meshing module and numerical module. More details of the three modules will be described later. The third layer is the graphical user interface including user-computer interactions. Currently, VCMM runs on Microsoft Windows and Linux systems.

The main interface window of VCMM is shown in Fig. 2. There are several regions at the user interface including the Menu Bar on the top, the Left Panel on the left side, the Status Bar at the bottom and the Render Area in the middle. These regions are described as below.

- *The Menu Bar*: The menu bar provides access to functions such as opening files, managing data, starting modules and invoking plug-ins.
- *Left Panel*: The left panel is a data manager. A data list and several buttons can be handled by users to manage memory.
- *Render Area*: The Render Area is where the 2D/3D representation of the scene is rendered. Mouse and keyboard interactions are provided in this area.
- *Status Bar*: There are some tips on the Status Bar including error messages, help messages and other information.

3. Modules

3.1. Molecular module

Molecular visualization is a main part of the molecular module. VCMM reads in a molecule coordinate file from the data manager and interactively displays the molecule on the screen in a variety of color schemes and molecule representations. VCMM supports most of the common representations for molecular structures: ball-and-stick, spheres, wire bonds, van der Walls surface and so on. When a user selects the sphere style (Fig. 3(b)), VCMM will adaptively control the resolution of each atom if the molecule has a large number of atoms. In this style, VCMM can display more than one million atoms on a typical PC in 2013.

Continuum modeling requires accurate and complete structural data as well as force field parameters such as atomic charges and radii, which information can be saved in a so called PQR file. In some cases, the molecular structure from PDB does not contain hydrogen atoms, and may even miss a fraction of the heavy atom coordinates. PDB2PQR [20] can provide the force field parameters and add the hydrogen atom and some heavy atoms missed in PDB. VCMM integrates the PDB2PQR package and enriches the database to handle some unusual amino acid types and ion species.

3.2. Meshing module

Meshing module includes mesh visualization, mesh generation and mesh quality analysis. Molecular surface meshing and volume meshing are useful in boundary/finite element modeling of biomolecules. The commonly used meshes are surface triangular mesh and volume tetrahedral mesh.

VCMM is able to render a given mesh in different styles as shown in Fig. 4. A mesh can be displayed as a surface or wire-frame by pressing the shortcut key like in Fig. 4(a) and (b). A surface mesh can be displayed directly, but the interior regions of a 3D volume mesh are difficult to render. In order to overcome this difficulty, VCMM provides some mesh filters to get a part of the mesh as shown in Fig. 4(c) and (d). Plane cutting is the most useful operation and it maintains the integrity of each volume cell. Users are also able to get a part of the mesh by selecting region marks as in Fig. 4(c).

If users have no molecular surface mesh file for input, VCMM can be used to generate it with the integrated mesh generation tools. VCMM contains a tool developed by our group, TMSmesh [21,22], for surface meshing on a molecular Gaussian surface. A Gaussian surface is defined as a level set of the summation of the Gaussian kernel functions. TMSmesh has a linear time complexity with respect to the number of atoms and is capable to handle arbitrarily large molecules (like a virus with more than 1 million atoms) of protein in PDB. In addition, MSMS [23] is another surface mesh generation tool as a plug-in in VCMM. MSMS is widely used for molecular surface triangulation for not huge molecules because of its high efficiency and relatively good quality. Once a surface mesh is obtained, a user can generate a volume mesh conforming to the molecular surface. The volume mesh can be generated

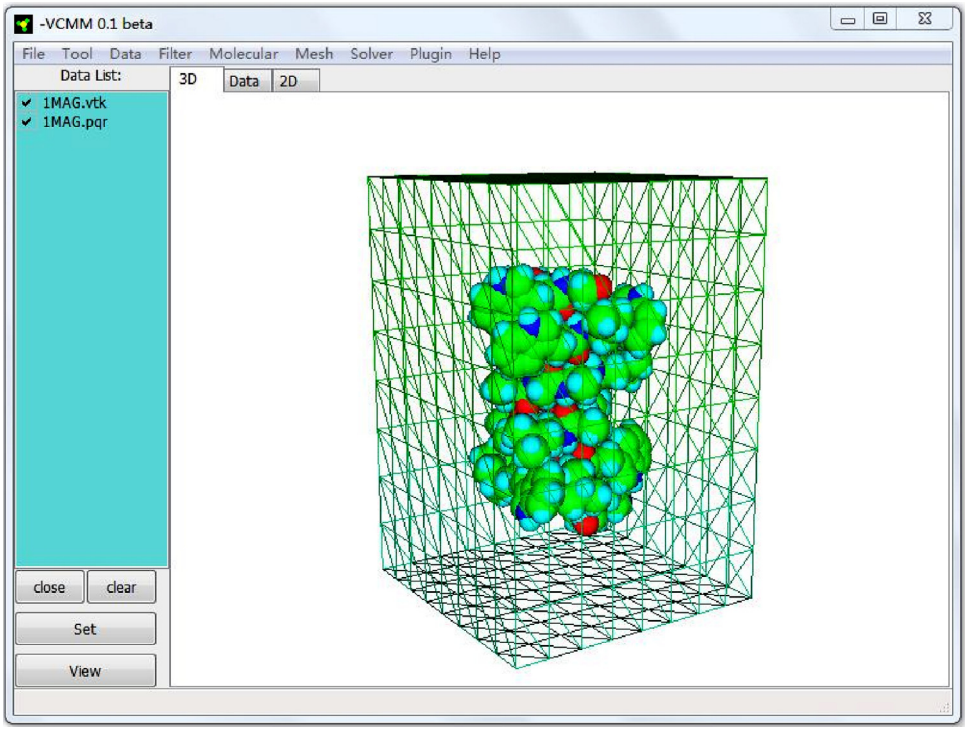

Fig. 2. VCMM main window.

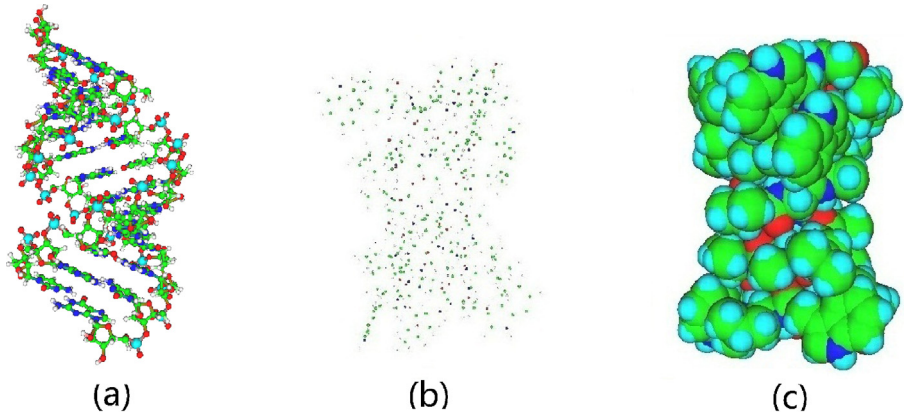

Fig. 3. Molecular representations of VCMM. (a) Ball-and-stick model, (b) sphere style, (c) Van der Walls surface.

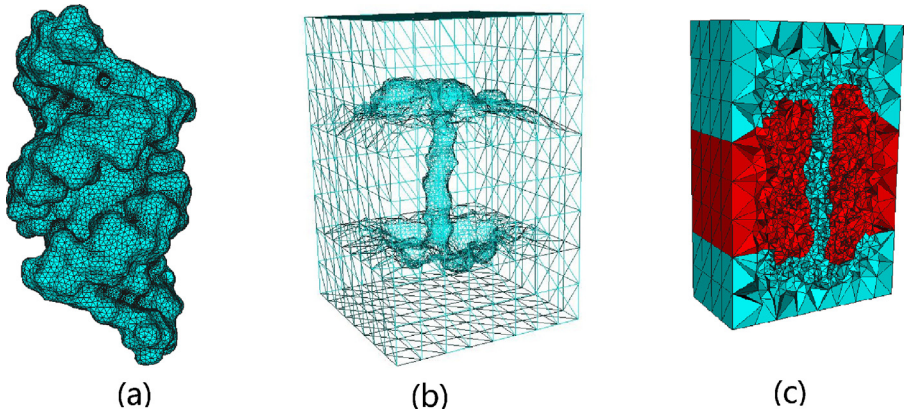

Fig. 4. Mesh conforming to an ion channel protein surface. (a) Surface mesh, (b) wire-frame of the volume mesh, (c) a cross section of the volume mesh with region marks.

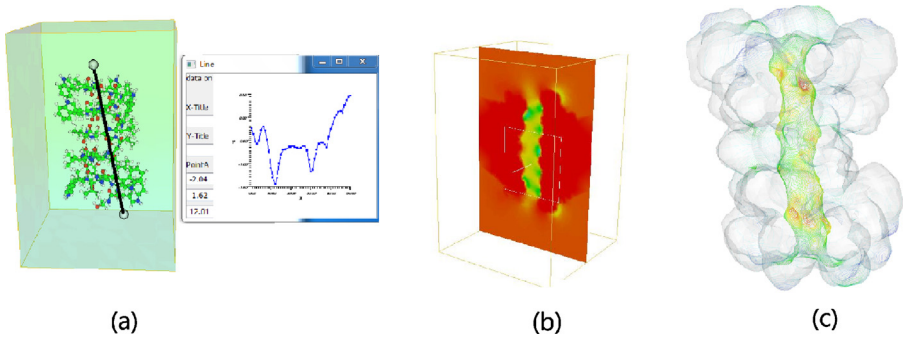

Fig. 5. Example of ion density distribution from an ion channel simulation. (a) Line interpolation, (b) a cut plane, (c) contour line on molecular surface.

by using TetGen [24]. TetGen is a software package to generate tetrahedral meshes of 3D polyhedral domains. It generates exact constrained Delaunay tetrahedralizations, boundary conforming Delaunay meshes, and Voronoi partitions. VCMM help users run TetGen in the graphical interface and the resulted mesh can be loaded into the data manager automatically.

Space management is the foundation of mesh management and visualization. It is difficult to retrieve the data from a given point in unstructured mesh directly. To find the cells around a given point, we must build a tree to locate each cell in a mesh. The octree and KD-tree are two widely adopted methods to build the cell tree. However, because a biomolecular object has a highly irregular shape and boundary, octree method wastes a lot of memory and the ordinary KD-tree method spends much time on pretreatment. A recent work on molecular mesh generation employed a KD-tree method to search atoms around a point [25]. Another fast and memory efficient cell location celltree method as described in [26] has been developed to manage unstructured mesh with general irregular objects embedded. This method is adopted by VCMM. However, the method is not specifically designed for molecular mesh. We improved the celltree method in VCMM by further taking into account the atomic position information. The molecular structure information is used to estimate the distribution of mesh point in order to divide a large mesh into several sub-meshes. Several top levels are constructed from the tertiary structure of proteins. The technique will be further improved, which will be much useful for speedup large molecular visualization, and offer VCMM a great advantage in space management.

3.3. *Numerical module*

Numerical solvers and visual analysis of numerical results are other two key parts of VCMM. An interface of numerical solvers helps users add finite/boundary element solvers into VCMM. VCMM can display numerical results and enable visual analysis.

VCMM supports triangular surface mesh and tetrahedral volume mesh. A solver can be interfaced to VCMM if it is based on these two types of meshes. To interface a solver to VCMM, the users need to write a python interface file which includes the input data format and parameters for the solver. A solver developed by our group is preloaded in VCMM, which is AFMPB, an adaptive fast multipole Poisson–Boltzmann solver for solving the linearized PB equation. Because of using the boundary integral method, AFMPB only needs the PQR file and a surface mesh of a molecule. The surface meshes generated by TMSmesh or MSMS are all accepted by AFMPB. Another solver *ichannel* [12] is under development, which uses a parallel finite element method for solving the 3D PNP equations for the ion channel system. The modeled channel system can consist of protein, membrane surrounding it, and a simulation box.

Numerical solution results are usually displayed as colored 3D objects. Users can look up the value by a color bar. Because it is impossible to observe the data inside a 3D model directly, VCMM visualizes the outside surface of the data set by default. A method to watch inside a volume data set is rendering a sub-region. A list of these sub-region renderings are illustrated as follows:

1 Interpolation on a straight line as shown in Fig. 5(a).
2 Interpolation on a plane as shown in Fig. 5(b).
3 Interpolation on a given surface as shown in Fig. 7.
4 Extraction contours and isosurface as show in Fig. 5(c).

Another method is volume rendering which has a set of techniques used to display a 3D data set. In scientific visualization, the technique of volume ray casting is the most popular one. VCMM uses volume ray casting algorithms and yields a high image quality as shown in Fig. 6. However, to display an unstructured mesh, volume ray casting is too slow for a usual computer. We develop an adaptive algorithm based on isosurface to describe a whole 3D data set in VCMM. The method is much faster than ray casting, but the produced image is of relatively low quality. The method details will be described in the manual or in future publications.

4. An example procedure to use VCMM

We select an ion channel protein gramicidin A (gA), which forms aqueous pores in lipid bilayers that is selectively permeable to monovalent cations. In order to calculate the surface electric potential of gA, we use MSMS to generate a molecular surface mesh and use AFMPB to solve the linear PBE. Firstly, we get a PDB file (PDB code: 1MAG) and load it into VCMM. VCMM will display the ball-and-stick style in the render window. Users can click labels in the left panel and select force field options on Molecular menu. Then,

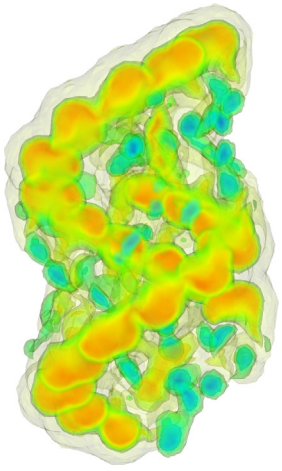

Fig. 6. Volume rendering of electrostatic potential around a DNA fragment.

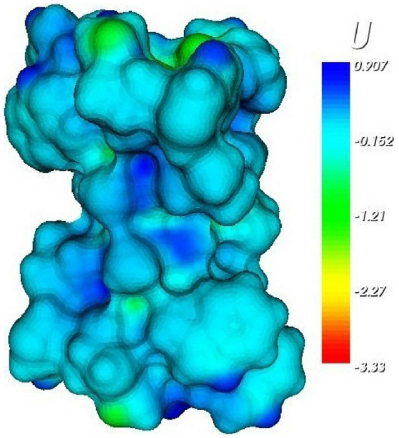

Fig. 7. Surface electric potential (in kcal/mol.e) resulted from the PB solver AFMPB on a gA channel protein.

a PQR file can be produced from the PDB file by using the tool PDB2PQR and choosing one of four commonly used force fields integrated in VCMM. Secondly, run the surface mesh generation plug-in MSMS and set the controlling parameters as "3, 1.4", which means the mesh resolution is 3 elements per Å^2 and the radius of the probing ball is 1.4 Å. When MSMS has been executed, a triangular surface mesh will be generated. Thirdly, load the PQR file and surface mesh into AFMPB and run the solver. The numerical result (surface potential) is displayed as in Fig. 7.

It is worth noting that VCMM enables a user to generate his/her own plug-ins. To this end, a user needs to create a Python script file in the directory of plug-in. VCMM will identify the python scripts and list them in the plug-in menu. When the user selects a plug-in, the plug-in script will be executed. The plug-in module grants users read and write access to the memory of VCMM. There are more details and examples of plug-in on our website.

The molecular surface mesh shown in Fig. 7 generated by a MSMS plug-in contains 4824 nodes and 9640 triangles. The total CPU time of this example is about 40 s on a Intel(R) Xeon(R) CPU E5620 2.4 GHz. With such a surface mesh, VCMM has rendered a high resolution surface electrostatic potential image which is calculated by AFMPB.

More details on the usage and performance of TMSmesh, MSMS, AFMPB and other possible plug-in software please refer to the corresponding references.

5. Discussion

This paper presents the design of VCMM, a visual and analysis tool written in Python. VCMM provides a graphical user interface and three modules to facilitate continuum molecular simulation. Meshing tools and numerical solvers can be invoked by VCMM. VCMM contains rich 3D scientific visualization tools for each part of the simulation. A model-based method for mesh management is developed, which can speed up the mesh indexing by using molecular structure. This method will enhance the capability of molecular visualization tools to handle unstructured mesh which is wildly used in finite/boundary element methods. In addition, we devote to develop more efficient visualization methods for the numerical results. The tools developed by our group, including VCMM, meshing tools and solvers, will gradually form a visual computing platform and provide facilities for finite/boundary element simulations. We are also developing solvers and plug-ins for PNP modeling of ion channels and nanopores [12]. This plug-in is to provide channel property analysis tools for the study of ion transport in the channel and nanopores. VCMM can also be useful in other fields such as computational quantum chemistry, material sciences, image processing, computational geometry and design, fluid simulation, and so on. Currently, VCMM only supports a few data formats and only handle visualization of linear elements. We will extend the functions to support more data formats such as XML and DX, and other mesh types such as regular grid (finite difference), hexahedral meshes, and high order element such as quadratic elements. The functionality for molecular visualization, other solvers and data analysis tools will be enriched as well in future releases.

6. Availability

VCMM is available free of charge for noncommercial use. Currently VCMM runs on Microsoft Windows and Linux systems. The binaries and plug-ins of VCMM can be obtained by accessing our web site https://www.continuummodel.org.

Acknowledgements

We would like to thank Bin Tu and other members in our group for their help. The work was supported by the State Key Laboratory of Scientific/Engineering Computing, National Center for Mathematics and Interdisciplinary Sciences, the Chinese Academy of Sciences and the China NSF (91230106), National 863 Project of China (2012AA020403).

References

[1] N.A. Baker, Improving implicit solvent simulations: a Poisson-centric view, Curr. Opin. Struct. Biol. 15 (2) (2005) 137–143.
[2] B.Z. Lu, Y.C. Zhou, M.J. Holst, J.A. McCammon, Recent progress in numerical methods for the Poisson–Boltzmann equation in biophysical applications, Commun. Comput. Phys. 3 (5) (2008) 973–1009.
[3] J.P. Bardhan, Biomolecular electrostatics – I want your solvation (model), Comput. Sci. Disc. 5 (1) (2012) 013001 http://stacks.iop.org/1749-4699/5/i=1/a=013001
[4] B.Z. Lu, Y.C. Zhou, G.A. Huber, S.D. Bond, M.J. Holst, J.A. McCammon, Electrodiffusion: a continuum modeling framework for biomolecular systems with realistic spatiotemporal resolution, J. Chem. Phys. 127 (13) (2007) 135102.
[5] C. Bajaj, S.C. Chen, A. Rand, An efficient higher-order fast multipole boundary element solution for Poisson–Boltzmann-based molecular electrostatics, SIAM J. Sci. Comput. 33 (2) (2011) 826–848.
[6] A.H. Boschitsch, M.O. Fenley, H.X. Zhou, Fast boundary element method for the linear Poisson–Boltzmann equation, J. Phys. Chem. B 106 (10) (2002) 2741–2754.
[7] B.Z. Lu, X.L. Cheng, J.F. Huang, J.A. McCammon, AFMPB: an adaptive fast multipole Poisson–Boltzmann Solver for calculating electrostatics in biomolecular systems, Comput. Phys. Commun. 181 (6) (2010) 1150–1160, http://dx.doi.org/10.1016/j.cpc.2010.02.015.
[8] R. Yokota, J.P. Bardhan, M.G. Knepley, L.A. Barba, T. Hamada, Biomolecular electrostatics using a fast multipole BEM on up to 512 GPUs and a billion unknowns, Comput. Phys. Commun. 182 (6) (2011) 1272–1283.
[9] W. H. Geng, F. Jacob, A GPU-accelerated direct-sum boundary integral Poisson–Boltzmann solver, Comput. Phys. Commun.
[10] M.J. Holst, F. Saied, Numerical solution of the nonlinear Poisson–Boltzmann equation: developing more robust and efficient methods, J. Comput. Chem. 16 (3) (1995) 337–364.
[11] D.X. Xie, Y. Jiang, P. Brune, L.R. Scott, A fast solver for a nonlocal dielectric continuum model, SIAM J. Sci. Comput. 34 (2) (2012) B107–B126.
[12] B. Tu, M.X. Chen, Y. Xie, L.B. Zhang, B. Eisenberg, B.Z. Lu, A parallel finite element simulator for ion transport through three-dimensional ion channel systems, J. Comput. Chem. 34 (24) (2013) 2065–2078, http://dx.doi.org/10.1002/jcc.23329.
[13] P.M. Kekenes-Huskey, A. Gillette, J. Hake, J.A. McCammon, Finite-element estimation of protein-ligand association rates with post-encounter effects: applications to calcium binding in troponin C and SERCA, Comput. Sci. Disc. 5 (1) (2012) 014015.
[14] W. L. DeLano, The PyMOL molecular graphics system, http://www.pymol.org/
[15] W. Humphrey, A. Dalke, K. Schulten, VMD: visual molecular dynamics, J. Mol. Graphics 14 (1) (1996) 33–38.
[16] A. Nicholls, K.A. Sharp, B. Honig, Protein folding and association: insights from the interfacial and thermodynamic properties of hydrocarbons, Proteins: Struct., Funct., Bioinf. 11 (4) (1991) 281–296.
[17] W. Schroeder, K. Martin, B. Lorensen, The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics, Kitware, Clifton Park, New York, 2006.
[18] J.D. Hunter, Matplotlib: a 2D graphics environment, Comput. Sci. Eng. (2007) 90–95.

[19] H.M. Berman, J. Westbrook, Z.K. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne, The protein data bank, Nucleic Acids Res. 28 (1) (2000) 235–242.
[20] T.J. Dolinsky, J.E. Nielsen, J.A. McCammon, N.A. Baker, PDB2PQR: an automated pipeline for the setup of Poisson–Boltzmann electrostatics calculations, Nucleic Acids Res. 32 (suppl. 2) (2004) W665–W667.
[21] M.X. Chen, B.Z. Lu, TMSmesh: a robust method for molecular surface mesh generation using a trace technique, J. Chem. Theory Comput. 7 (1) (2010) 203–212.
[22] M.X. Chen, B. Tu, B. Lu, Triangulated manifold meshing method preserving molecular surface topology, J. Mol. Graphics Modell. 38 (2012) 411–418, http://dx.doi.org/10.1016/j.jmgm.2012.09.006.
[23] M.F. Sanner, A.J. Olson, J.C. Spehner, Reduced surface: an efficient way to compute molecular surfaces, Biopolymers 38 (3) (1996) 305–320.
[24] H. Si, TetGen. A quality tetrahedral mesh generator and a 3D Delaunay triangulator, http://tetgen.berlios.de/
[25] T. Liao, Y. Zhang, P.M. Kekenes-Huskey, Y. Cheng, A. Michailova, A.D. McCulloch, M. Holst, J.A. McCammon, Multi-core CPU or GPU-accelerated multiscale modeling for biomolecular complexes, Mol. Based Math. Biol. 1 (2013) 164–179.
[26] C. Garth, K.I. Joy, Fast, memory-efficient cell location in unstructured grids for visualization, IEEE Trans. Visual. Comput. Graphics 16 (6) (2010) 1541–1550.