Journal of Computational Physics 448 (2022) 110718

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Letter to the editor

Efficient calculation of fully resolved electrostatics around large biomolecules

Rochishnu Chowdhury [a,*], Raphael Egan [a], Daniil Bochkov [a], Frederic Gibou [a,b]

[a] *Department of Mechanical Engineering, University of California, Santa Barbara, CA 93106-5070, United States of America*
[b] *Department of Computer Science, University of California, Santa Barbara, CA 93106-5110, United States of America*

ARTICLE INFO

Article history:
Available online 29 September 2021

Keywords:
Nonlinear Poisson-Boltzmann equation
Immersed interface
Level-set method
Solvation energy
Biomolecules

ABSTRACT

We present a simple framework for calculating the electric potential by solving the nonlinear Poisson-Boltzmann equation and the free solvation energies of large biomolecules. To achieve this we build upon the work of Bochkov and Gibou [9] to develop a novel solver capable of solving nonlinear elliptic equations, where the diffusion coefficient, the source term, the solution and its flux are discontinuous across the interface. The interface is represented by the zero-level set of a signed distance function, empowering a natural and systematic approach to generate adaptive Cartesian grids, which drastically reduce the computational cost by focusing resources to regions near the surface of the molecules. The solver is implemented on a forest of Octree grids in parallel to enable fast computations over large molecules.

Published by Elsevier Inc.

1. Introduction

The Poisson-Boltzmann (PB) theory is well established in many scientific fields, e.g. electrochemistry (Guoy-Chapman theory [35,14]), solution chemistry (Debye-Hückel theory) [20], colloid chemistry (Derjaguin-Landau-Verwey-Overbeek theory) [21,73] and biophysics (PB theory) [19,39]. The Poisson-Boltzmann equation (PBE) is an implicit solvent model that considers a microscopic treatment of the solute and characterizes the solvent using its macroscopic physical properties (i.e. the solvent dielectric permittivity, the ionic strength, etc.) as opposed to the description of explicit solvent models that are based on a molecular representation of both the solute and the solvent molecules. Explicit solvent models produce accurate results but their computational cost quickly becomes prohibitive for larger molecules. Implicit solvent models are less accurate but are significantly more computationally efficient and they have been shown to produce satisfactory results when computing electric fields and solvation energies, see e.g. [32,39,67,68].

Several advances have been made to solve the PBE numerically. Approaches include Finite Difference Methods (FDM) [13,75,53,45,16,4,40,64,6,58], Finite Element Methods (FEM) [3,37,38,10,15], Boundary Element Methods (BEM) [5,41,48,65,74,78] and quantum or hybrid approaches [15,49–51,66]. We highlight some of the main characteristics of these techniques.

The BEM is a very efficient approach to solve the linear Poisson-Boltzmann equation (LPBE); however this model is only accurate when the ionic strength approaches zero, which is in contradiction with typical biochemical applications that involve relatively high ionic strengths. The main disadvantage of this method is that it cannot be easily generalized to solve the nonlinear Poisson-Boltzmann equation (NPBE); authors have nevertheless introduced hybrid models leveraging

[*] Corresponding author.
E-mail address: rochishnu00@ucsb.edu (R. Chowdhury).

https://doi.org/10.1016/j.jcp.2021.110718
0021-9991/Published by Elsevier Inc.

its strengths with those of the FEM formulations [11,12]. The ability to enforce continuity conditions at the interface and the natural capability for adaptive refinement with a priori error estimates had given Finite Element/Volume Methods an advantage over then existing Finite Difference Methods. In this framework, it is undemanding to impose the continuity condition but grid generation suffers in cases of large deformations. Implicit representation of surface of a molecule is fast and robust but imposing jump conditions on such a framework is challenging. Finite Difference Methods are typically used in the context of implicit surface representations.

Imposing sharp jump conditions was first attempted with the advent of the Immersed Interface Method (IIM) for the Poisson equation [44], where the jump conditions are combined with Taylor expansions of the solution on each side of the interface to modify the numerical stencils at the grid points close to the interface. The difficulties with this approach include calculating higher-order derivatives and jump conditions. Several subsequent research works have led to several improvements of the IIM framework [18,46,47,76,7,1,2,72] and the references therein. The philosophy of the Ghost Fluid Method (GFM) [26], initially introduced to treat shocks and contact discontinuities in compressible flows was used for solving the Poisson equation with jump conditions in Liu et al. [52]. The projection of the jump in the normal flux of the solution onto the Cartesian directions enables the treatment of arbitrary smooth interfaces in a dimension-by-dimension approach in two and three spatial dimensions. However, this approach neglects jumps in tangential components, which introduces an inconsistency and produces inaccurate gradients close to the interface as shown in Egan et al. [25]. One way of tackling this problem is introduced in the Voronoi Interface Method [33], which considers the GFM treatment after constructing a local Voronoi mesh adjacent to the interface; this method has been successfully applied to electroporation problems [34,59]. However, the generation of local interface-conforming Voronoi mesh is challenging, especially in 3D, and brings the major computational challenge of mesh construction back into play. Later in the paper [25], the first treatment that preserves the advantages of the GFM (symmetric definite positive linear system on Cartesian grids) while producing convergent fluxes in the maximum norm, was presented. Another reliable and robust approach was recently proposed by Bochkov et al. [9], where the jump conditions were taken into account by adopting the idea of using Taylor expansions in the normal direction and employing local least-square interpolations. This technique exploits a Finite Volume formulation with two unknown values, the local value of the solution and a corresponding ghost value for grid nodes whose control volume is crossed by the interface; the ghost values are defined by means of Taylor expansions in the interface-normal direction, constrained to satisfy the interface jump conditions at the projection of the considered grid node onto the interface.

Initially FDMs were restricted to uniform grids, the grid points were marked solvent accessible or inaccessible, and the same discretization was applied everywhere regardless of the conditions at the solute-solvent interface. The major advantage is the ease of implementation, but this comes at the expense of accuracy because the quantities across the interface that must be described as discontinuous at the macroscopic level are numerically smeared. Recent developments in Finite Difference techniques have addressed this issue [33,79]. In addition, high resolution grids can be considered in this framework thanks to advanced numerical methods on adaptive grids [36,57,58].

In the present work, we describe an efficient framework for solving the NPBE to calculate the electric potential around large biomolecules and their solvation free energies. The key component of this approach is a solver of nonlinear elliptic equations capable of dealing with interface discontinuities in the diffusion coefficient, the source term, the solution and its flux across the interface developed in section 3. There are multiple ways of defining a molecular surface; we use the Solvent-Excluded Surface (SES) constructed using the fast and scalable tool developed in [24] for its capability of considering large molecules. To test our approach, we first consider the Kirkwood model, which has an explicit analytical solution and then we compare the solvation energies of different biomolecules generated by our solver with the values calculated using the popular open-source software, APBS [42,4]. The scalability of the solver is also demonstrated.

2. The nonlinear Poisson-Boltzmann equation

The NPBE for symmetric $1:1$ electrolytes is given by

$$-\tilde{\nabla}\cdot\left(\tilde{\varepsilon}_{\circ}\varepsilon_{\mathrm{r}}\tilde{\nabla}\tilde{\psi}\right)+2z\tilde{e}\tilde{n}_{\infty}\sinh\left(\frac{z\tilde{e}\tilde{\psi}}{\tilde{k}_{\mathrm{B}}\tilde{T}}\right)=\sum_{k=1}^{N_{\mathrm{at.}}}q_{k}\tilde{e}\tilde{\delta}\left(\tilde{\mathbf{x}}-\tilde{\mathbf{x}}_{k}\right), \tag{1}$$

where $\tilde{\varepsilon}_{\circ}$ is the absolute dielectric permittivity of classical vacuum and ε_{r} is the relative permittivity of the solvent, z is the ionic charge of the ions present in the solvent in units of the elementary charge $\tilde{e}$, $\tilde{n}_{\infty}$ is the far-field ion density, $\tilde{k}_{\mathrm{B}}$ is the Boltzmann constant and $\tilde{T}$ is the temperature. The solute contains point-distributed charges of (total) charge q_k located at $\tilde{\mathbf{x}}_k$ (where k goes from 1 to $N_{\mathrm{at.}}$, $N_{\mathrm{at.}}$ being the total number of atoms in the solute). Details about derivation of the NPBE are given in [69]. The presence of tilde above the quantities indicates that the quantities are dimensional. The solute is described by the negative subdomain (Ω^{-}) and the solvent is described by the positive subdomain (Ω^{+}) separated by the interface (Γ) as described by Fig. 1. By defining the non-dimensional electrostatic potential as $\psi=\dfrac{z\tilde{e}}{\tilde{k}_{\mathrm{B}}\tilde{T}}\tilde{\psi}$ and a normalization length $\tilde{\ell}$, we can rewrite (1) as

$$-\nabla\cdot\left(\varepsilon_{\mathrm{r}}\nabla\psi\right)+\left(\kappa\left(\mathbf{x}\right)\right)^{2}\sinh(\psi)=\sum_{k=1}^{N_{\mathrm{at.}}}\frac{q_{k}\tilde{e}^{2}z}{\tilde{\varepsilon}_{\circ}\tilde{k}_{\mathrm{B}}\tilde{T}\tilde{\ell}}\delta\left(\mathbf{x}-\mathbf{x}_{k}\right), \tag{2}$$

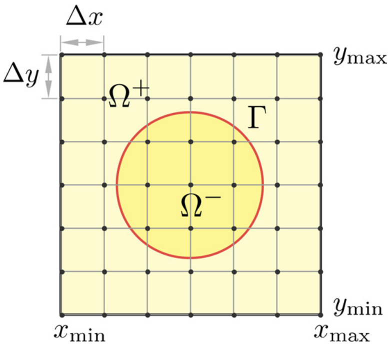


Fig. 1. Notation used in the paper.

where the nonlinear factor $\kappa\,(\mathbf{x})$ is a piece-wise constant which is zero in the solute and $\frac{\tilde{\ell}}{\tilde{\lambda}}$ ($\tilde{\lambda}$ is the modified Debye Length and $\tilde{\lambda}^2 = \frac{\tilde{\varepsilon}_\circ \tilde{k}_\mathrm{B} \tilde{T}}{2z^2 \tilde{e}^2 \tilde{n}_\infty}$) in the solvent.

The physical boundary condition to consider in such a context is a far-field boundary condition,

$$\lim_{\|\mathbf{x}\| \to \infty} \psi\,(\mathbf{x}) = 0. \tag{3}$$

However, we are limited to finite-domain calculations and the development of appropriate technique(s) to mimic the above far-field boundary condition in a finite domain calculation is beyond the scope of this work. We opt for a simpler Dirichlet boundary condition

$$\psi = f\,(\mathbf{x})\,, \quad \forall \mathbf{x} \in \partial\Omega, \tag{4}$$

where f is a function prescribed by the user. Appropriate modeling choices for f exist to mimic (3) as best as possible, especially in case of linearized Poisson-Boltzmann problems [66]. For actual molecular compounds, we use $f = 0$ with a relatively large domain compared to the size of the compound so that the numerical effects of a finite size domain are negligible. The interface boundary conditions are

$$[\psi] = 0 \textit{ and } [\varepsilon_\mathrm{r} \nabla \psi \cdot \mathbf{n}] = 0. \tag{5}$$

The presence of point-distributed charges in the solvent creates certain difficulties which can be alleviated by making use of the analytical form

$$\psi^\star\,(\mathbf{x}) = \sum_{k=1}^{N_\mathrm{at.}} \left(\frac{q_k \tilde{e}^2 z}{\tilde{\varepsilon}_\circ \tilde{k}_\mathrm{B} \tilde{T} \tilde{\ell}} \right) \frac{1}{4\pi\,\varepsilon_\mathrm{r,m} \left\| \mathbf{x} - \mathbf{x}_k \right\|}, \tag{6}$$

which satisfies $-\nabla \cdot \left(\tilde{\varepsilon}_\circ \varepsilon_\mathrm{r,m} \nabla \psi^\star \right) = \sum_{k=1}^{N_\mathrm{at.}} \frac{q_k \tilde{e}^2 z}{\tilde{k}_\mathrm{B} \tilde{T} \tilde{\ell}} \delta\,(\mathbf{x} - \mathbf{x}_k)$ in $\mathbb{R}^3$ as discussed in [17,80] and [28], where $\varepsilon_\mathrm{r,m}$ is the relative permittivity of the molecule. The non-dimensional electric potential ψ is now defined as

$$\psi = \begin{cases} \check{\psi} & \text{in } \Omega^+, \\ \check{\psi} + \psi^\star & \text{in } \Omega^-. \end{cases} \tag{7}$$

Upon substituting (7) into (2) and exploiting the property of the analytical form $\psi^\star$, we have[1]

$$\begin{cases} -\nabla \cdot \left(\varepsilon_\mathrm{r} \nabla \check{\psi} \right) + \left(\kappa\,(\mathbf{x}) \right)^2 \sinh\!\left(\check{\psi} \right) = 0, & \\ \qquad\qquad \check{\psi} \quad = 0 & \text{on } \partial\Omega, \\ \qquad\quad\; \left[\check{\psi} \right] \quad = +\psi^\star & \text{across } \Gamma, \\ \left[\varepsilon_\mathrm{r} \nabla \check{\psi} \cdot \mathbf{n} \right] \quad = \varepsilon_\mathrm{r,m} \nabla \psi^\star \cdot \mathbf{n} & \text{across } \Gamma. \end{cases} \tag{8}$$

The above equation is solved using the nonlinear elliptic solver described in Section 3. Once ψ is known everywhere, the free solvation energy of the biomolecule is calculated using

$$\mathcal{E}^\mathrm{s} = \tilde{k}_\mathrm{B} \tilde{T} \sum_{k=1}^{N_\mathrm{at.}} \frac{q_k}{2z} \left(\check{\psi}\,(\mathbf{x}_k) \right) + \tilde{k}_\mathrm{B} \tilde{T} \tilde{n}_\infty \tilde{\ell}^3 \int_{\Omega^+} \left(\psi \sinh\,(\psi) - 2\,(\cosh\,(\psi) - 1) \right) \mathrm{d}\mathbf{x}, \tag{9}$$

[1] Note that $\kappa \neq 0$ only where $\psi = \check{\psi}$ by construction.

as discussed in [54].

3. Nonlinear elliptic solver

We consider the nonlinear elliptic equations of the form:

$$-\nabla \cdot (\mu(\mathbf{x})\nabla u(\mathbf{x})) + w(\mathbf{x})u(\mathbf{x}) + k(\mathbf{x})p(u(\mathbf{x})) = f(\mathbf{x}) \text{ in } \Omega \subset \mathbb{R}^{\mathrm{d}}, \tag{10}$$

where d is the problem dimensionality. The scalar functions $\mu(\mathbf{x})$, $w(\mathbf{x})$, $k(\mathbf{x})$, $u(\mathbf{x})$), $f(\mathbf{x})$ and the nonlinear function $p(u(\mathbf{x}))$ may be only piecewise continuous, and may have jumps across internal interfaces in the domain. Although we focus on the nonlinear Poisson-Boltzmann equation for solving for the electric field in the context of biophysics applications, equations of the form (10) appear in various other applications like flows in porous media, population genetics, astrophysics, combustion, semiconductor physics and biophysics.

3.1. Numerical discretization

We present the numerical discretizations in two spatial dimensions, noting that the extension to three-dimensional cases is straightforward. Consider a rectangular domain $\Omega = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$ with an immersed irregular interface Γ that splits Ω into two sets Ω^- and Ω^+. We seek a numerical solution $u = u(\mathbf{x})$, with $\mathbf{x} = (x, y)$, to the following problem:

$$w^{\pm}u^{\pm} + k^{\pm}p^{\pm} - \nabla \cdot \left(\mu^{\pm}\nabla u^{\pm}\right) = f^{\pm}, \quad \text{in} \quad \Omega^{\pm}, \tag{11}$$

$$\begin{aligned} [u] &= \alpha, \quad \text{on} \quad \Gamma, \\ [\mu\partial_n u] &= \beta, \quad \text{on} \quad \Gamma, \end{aligned} \tag{12}$$

$$u^{+} = g, \quad \text{on} \quad \partial\Omega, \tag{13}$$

where the functions $w^{\pm} = w^{\pm}(\mathbf{x})$, $k^{\pm} = k^{\pm}(\mathbf{x})$, $\mu^{\pm} = \mu^{\pm}(\mathbf{x}) \geq \epsilon > 0$, $f^{\pm} = f^{\pm}(\mathbf{x})$, and the nonlinear function $p^{\pm} = p^{\pm}(u)$ and its derivative $p^{\prime,\pm} = p^{\prime,\pm}(u)$. For simplicity we impose Dirichlet boundary conditions on the boundary of the computation domain, i.e. (13) and we assume $\Gamma \bigcap \partial\Omega = \emptyset$. We discretize the domain Ω into a grid of $N_x \times N_y$ points with spatial steps

$$\Delta x = \frac{x_{\max} - x_{\min}}{N_x - 1}, \quad \Delta y = \frac{y_{\max} - y_{\min}}{N_y - 1},$$

and associate with each point $\mathbf{x}_{i,j} = \left(x_i, y_j\right) = (x_{\min} + (i-1)\Delta x, y_{\min} + (j-1)\Delta y)$, a finite volume $V_{i,j} = \left[x_i - \frac{1}{2}\Delta x, x_i + \frac{1}{2}\Delta x\right] \times \left[y_j - \frac{1}{2}\Delta y, y_j + \frac{1}{2}\Delta y\right]$, $i \in [2, N_x - 1]$, $j \in \left[2, N_y - 1\right]$. The Level-Set Method [62,31] is used to describe the irregular interface Γ using a Lipschitz-continuous function $\phi(\mathbf{x})$ such that $\Omega^+ = \{\mathbf{x} : \phi(\mathbf{x}) > 0\}$, $\Omega^- = \{\mathbf{x} : \phi(\mathbf{x}) < 0\}$ and $\Gamma = \{\mathbf{x} : \phi(\mathbf{x}) = 0\}$, with $|\nabla\phi| = 1$. It is noted that even though a signed distance function is desired for robustness, it is only needed to avoid functions with very steep or flat gradients. The domain is discretized into squares that are represented on a Quadtree data structure. Fig. 2 (left) illustrates the computational domain in 2D along with its corresponding Quadtree (right). The generation of the grid is initiated by appointing the tree root, i.e. level zero, to the whole domain and recursively splitting every cell (level z) into four for 2D smaller cells (level z+1) until either a certain resolution criterion is met or the tree has reached its maximum level. Following Min et al. [56] and Strain [71], one such criterion may be chosen as to divide a cell, with vertices in the set V, if the following inequality holds true:

$$\min_{v \in V} |\phi(v)| < \frac{LD}{2}, \tag{14}$$

where L is the Lipschitz constant of the level set function ϕ and D is the diagonal size of the current cell. $L = 1.2$ for all our refinements. By definition, the tree will be graded if the level difference between any two adjacent cells is at most one and non-graded if there is no such restriction. In this paper we consider non-graded Cartesian grids where the solution is sampled at the nodes of each cell. A node in the grid is said to be uniform if it is directly connected to other nodes in each of six ± Cartesian directions and is said to have a T-junction if it does not have a direct neighbor in at least one of the six ± Cartesian directions.

The main challenge in developing numerical methods for adaptive Cartesian meshes is addressing T-junctions accurately. In [56], the authors showed that second-order accurate discretizations can be obtained on highly non-graded Cartesian meshes by counterpoising numerical error in one or two spatial directions with the derivative in the transverse direction. Furthermore, they showed that such a discretization is always possible in the case where the solution is sampled at the nodes of each cell. Far from the interface, a second-order accurate finite difference method is used to discretize (11), the details of the approach are given in [57]. On the boundaries, the Dirichlet boundary conditions are imposed by any of the

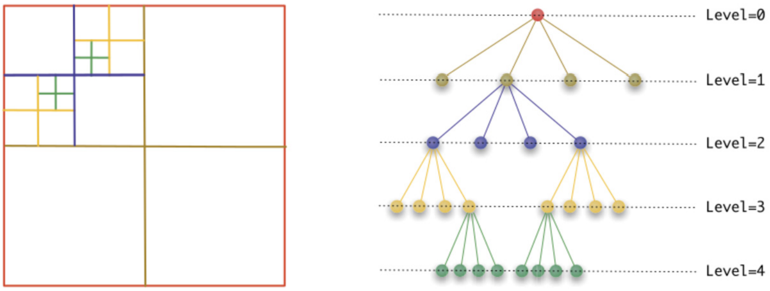
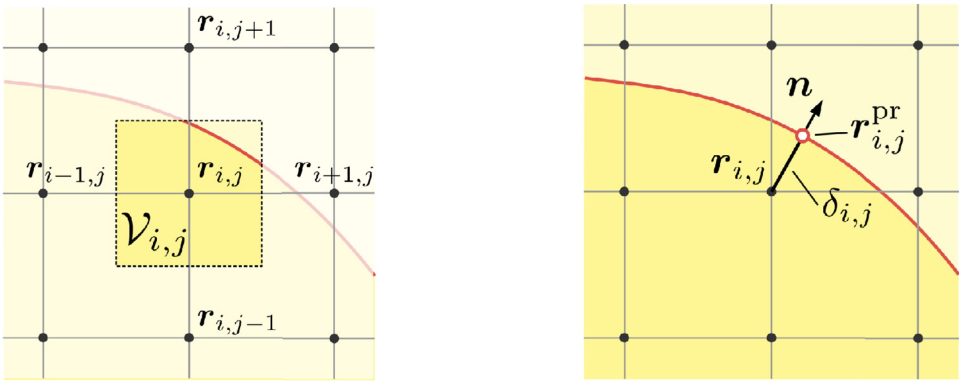


Fig. 2. Non-graded adaptive discretization of a two-dimensional domain (left) and its corresponding Quadtree (right).



(a)

(b)

Fig. 3. (a) Illustration of a finite volume associated with a grid point (i, j). (b) Illustration of a grid point projected onto the interface Γ.

methods [29,30,70]. Our main focus in this paper is to develop a Finite Volume approach to handle the jump conditions (12) in the NPBE, notations used next are presented in Fig. 3.

First, let us discuss the discretization of the linear part of the equation, i.e. when $k^{\pm} = 0$. Consider a point $\mathbf{x}_{i,j}$ such that its finite volume $V_{i,j}$ is crossed by Γ, integrating equation (11) over $V_{i,j}$ and applying the divergence theorem, the following expression is obtained:

$$\sum_{s=+,-}\int_{\Omega^s\cap V_{i,j}} w^s u^s \mathrm{d}\Omega - \sum_{s=+,-}\int_{\Omega^s\cap\partial V_{i,j}} \mu^s \partial_{n^s} u^s \mathrm{d}\Gamma = \int_{\Omega^s\cap V_{i,j}} f^s \mathrm{d}\Omega - \int_{\Gamma\cap V_{i,j}} [\mu\partial_n u]\,\mathrm{d}\Gamma. \tag{15}$$

Following [60,63], the domain integrals are approximated by the integrand value multiplied by the corresponding volumes and estimating fluxes between cells using values at nearest-neighbor grid points and central difference points, we get:

$$\begin{aligned}\sum_{s=+,-} w^s_{i,j}u^s_{i,j}|V^s_{i,j}| - \sum_{s=+,-}\ & \left\{\mu^s_{i-\frac{1}{2},j}A^s_{i-\frac{1}{2},j}\frac{u^s_{i-1,j}-u^s_{i,j}}{\Delta x} + \mu^s_{i+\frac{1}{2},j}A^s_{i+\frac{1}{2},j}\frac{u^s_{i+1,j}-u^s_{i,j}}{\Delta x}\right. \\ & \left. +\mu^s_{i,j-\frac{1}{2}}A^s_{i,j-\frac{1}{2}}\frac{u^s_{i,j-1}-u^s_{i,j}}{\Delta y} + \mu^s_{i,j+\frac{1}{2}}A^s_{i,j+\frac{1}{2}}\frac{u^s_{i,j+1}-u^s_{i,j}}{\Delta y}\right\} \\ & = \sum_{s=+,-} f^s_{i,j}|V^s_{i,j}| - \int_{\Gamma\cap V_{i,j}} \beta\mathrm{d}\Gamma + O\left(h^d\right),\end{aligned} \tag{16}$$

where d is the problem dimensionality, $h = \max(\Delta x, \Delta y)$, $|V^{\pm}_{i,j}|$ denotes the volume of $V_{i,j} \cap \Omega^{\pm}$, $u_{i,j} = u\left(\mathbf{x}_{i,j}\right)$, $A^{\pm}_{i\pm\frac{1}{2},j}$ and $A^{\pm}_{i,j\pm\frac{1}{2}}$ are face areas of $V^{\pm}_{i,j}$ in the x- and y-directions, respectively. To compute the boundary and domain integrals required by the proposed discretization, the geometric reconstruction approach from [55] is used. For an immersed interface that is only piece-wise smooth the method from [8] can be used.

The discretization given by equation (16) requires that both the values of u^- and u^+ be available at grid points with a control volume crossed by Γ. This strategy leads to more unknowns than equations and the consequent under-determination of the system is addressed by constraining these local couple of values via the relation:

$$[u]_{i,j} = u^+_{i,j} - u^-_{i,j} = \alpha + \delta_{i,j}\,[\partial_{\mathbf{n}}u]_{i,j}$$

$$= \alpha + \delta_{i,j} \frac{\beta - [\mu]\, \partial_{\mathbf{n}} u^-}{\mu^+} \tag{17}$$

$$\overset{\text{or}}{=} \alpha + \delta_{i,j} \frac{\beta - [\mu]\, \partial_{\mathbf{n}} u^+}{\mu^-} \tag{18}$$

where $\delta_{i,j}$ is the (signed) distance of the considered grid node (i, j) to Γ. The required value of either $\partial_{\mathbf{n}} u^-$ or $\partial_{\mathbf{n}} u^+$ is then evaluated as $\partial_{\mathbf{n}} u^{\pm} = \left(c^{\pm}_{i,j} u^{\pm}_{i,j} + \sum_{(p,q) \in N^{\pm}_{i,j}} c^{\pm}_{i,j,p,q} u^{\pm}_{i+p,j+q} \right) + O(h)$ (details about the coefficients $c^{\pm}_{i,j}$ and $c^{\pm}_{i,j,p,q}$ are present in [8]) from the local linear interpolant built to minimize the square of the errors with either $u^+_{i,j}$ or $u^-_{i,j}$ and the neighbor values of u at the surrounding nodes 3×3 neighborhood nodes centered on (i, j) that belong either to Ω^+ or Ω^- (these subsets are respectively denoted $N^+_{i,j}$ and $N^-_{i,j}$). This procedure effectively addresses the under-determination of the overall system since it leads to the following $\mathcal{O}\left(h^2\right)$ expressions:

- for $\mathbf{x}_{i,j} \in \Omega^-$

$$u^+_{i,j} = u^-_{i,j} + \alpha + \delta_{i,j} \left\{ \frac{\beta}{\mu^+} - \frac{[\mu]}{\mu^+} \left(c^-_{i,j} u^-_{i,j} + \sum_{(p,q) \in N^-_{i,j}} c^-_{i,j,p,q} u^-_{i+p,j+q} \right) \right\}$$

$$\overset{\text{or}}{=} u^-_{i,j} + \alpha + \delta_{i,j} \left\{ \frac{\beta}{\mu^-} - \frac{[\mu]}{\mu^-} \left(\frac{c^+_{i,j} \left(u^-_{i,j} + \alpha + \delta_{i,j} \frac{\beta}{\mu^-} \right) + \sum_{(p,q) \in N^+_{i,j}} c^+_{i,j,p,q} u^+_{i+p,j+q}}{1 + \delta_{i,j} \frac{[\mu]}{\mu^-} c^+_{i,j}} \right) \right\}$$

- for $\mathbf{x}_{i,j} \in \Omega^+$

$$u^-_{i,j} = u^+_{i,j} - \alpha - \delta_{i,j} \left\{ \frac{\beta}{\mu^+} - \frac{[\mu]}{\mu^+} \left(\frac{c^-_{i,j} \left(u^+_{i,j} - \alpha - \delta_{i,j} \frac{\beta}{\mu^+} \right) + \sum_{(p,q) \in N^-_{i,j}} c^-_{i,j,p,q} u^-_{i+p,j+q}}{1 - \delta_{i,j} \frac{[\mu]}{\mu^+} c^-_{i,j}} \right) \right\}$$

$$\overset{\text{or}}{=} u^+_{i,j} - \alpha - \delta_{i,j} \left\{ \frac{\beta}{\mu^-} - \frac{[\mu]}{\mu^-} \left(c^+_{i,j} u^+_{i,j} + \sum_{(p,q) \in N^+_{i,j}} c^+_{i,j,p,q} u^+_{i+p,j+q} \right) \right\}.$$

In turn, we have only $N_x \times N_y$ unknowns left, defined as

$$u_{i,j} = \begin{cases} u^+_{i,j}, & \mathbf{x}_{i,j} \in \Omega^+, \\ u^-_{i,j}, & \mathbf{x}_{i,j} \in \Omega^-. \end{cases} \tag{19}$$

The above rules contain two different formulas for eliminating the ghost degrees of freedom: one is based on approximating $\partial_{\mathbf{n}} u^-$ (first equations), while the other approximates $\partial_{\mathbf{n}} u^+$ (second equations). Therefore, there is a certain flexibility to construct the final discretization: it is shown in [8] that the most stable and best-conditioned scheme, referred to as `Bias Slow`, uses interpolation from the slow-diffusion region (e.g., if $\mu^- > \mu^+$ then we use the formula based on the estimation of $\partial_n u^+$).

When $k^{\pm}$ is nonzero, Newton's iteration method is used to treat the nonlinear term. Considering a series of solution u^l, and starting with an initial guess u^0, a Taylor series expansion for the nonlinear term is utilized to locally linearize $p^{\pm,l+1}$ around u^l in (11) as

$$p^{\pm,l+1} \approx p^{\pm,l} + p'^{,\pm,l} (u^{l+1} - u^l),$$

where $p^{\pm,l} = p^{\pm}(u^l)$. Using the linearization, (11) becomes:

$$w^{\pm} u^{\pm,l+1} + k^{\pm} p'^{,\pm,l} u^{\pm,l+1} - \nabla \cdot \left(\mu^{\pm} \nabla u^{\pm,l+1} \right) = f^{\pm} - k^{\pm} p^{\pm,l} + k^{\pm} p'^{,\pm,l} u^{\pm,l}. \tag{20}$$

Therefore, the discretization is

$$\begin{aligned}&\textstyle\sum_{s=+,-}\left\{w^s_{i,j}|V^s_{i,j}|+\frac{\mu^s_{i-\frac{1}{2},j}A^s_{i-\frac{1}{2},j}+\mu^s_{i+\frac{1}{2},j}A^s_{i+\frac{1}{2},j}}{\Delta x}+\frac{\mu^s_{i,j-\frac{1}{2}}A^s_{i,j-\frac{1}{2}}+\mu^s_{i,j+\frac{1}{2}}A^s_{i,j+\frac{1}{2}}}{\Delta y}+(\text{add to diag})^s\right\}u^{s,l+1}_{i,j}\\ &-\textstyle\sum_{s=+,-}\left\{\mu^s_{i-\frac{1}{2},j}A^s_{i-\frac{1}{2},j}\frac{u^{s,l+1}_{i-1,j}}{\Delta x}+\mu^s_{i+\frac{1}{2},j}A^s_{i+\frac{1}{2},j}\frac{u^{s,l+1}_{i+1,j}}{\Delta x}+\mu^s_{i,j-\frac{1}{2}}A^s_{i,j-\frac{1}{2}}\frac{u^{s,l+1}_{i,j-1}}{\Delta y}+\right.\\ &\left.\mu^s_{i,j+\frac{1}{2}}A^s_{i,j+\frac{1}{2}}\frac{u^{s,l+1}_{i,j+1}}{\Delta y}\right\}=\textstyle\sum_{s=+,-}f^s_{i,j}|V^s_{i,j}|-\int_{\Gamma\cap V_{i,j}}\beta d\Gamma+(\text{add to rhs})+O(h^d),\end{aligned}\tag{21}$$

where

$$(\text{add to diag})^s=k^s_{i,j}p'^{,s}\left(u^{s,l}_{i,j}\right)|V^s_{i,j}|\tag{22}$$

and

$$(\text{add to rhs})^s=-\sum_{s=+,-}k^s_{i,j}\left\{p^s(u^{s,l}_{i,j})-p'^{,l}\left(u^{s,l}_{i,j}\right)u^{s,l}_{i,j}\right\}|V^s_{i,j}|.\tag{23}$$

(add to diag) and (add to rhs) are updated at every iteration and the iteration scheme is carried out until the difference in the solution reaches a given tolerance or the residual of the nonlinear partial difference equation reaches a desired residual tolerance.

The linear system of equations formed by the numerical discretization (21) is denoted by

$$\mathbf{A}\mathbf{u}^{\pm,l+\mathbf{1}}=\mathbf{R}^l.\tag{24}$$

Instead of solving for the next iterate $\mathbf{u}^{\pm,l+\mathbf{1}}$, the increment of solution between two iterates $\delta\mathbf{u}^{\pm,l}$ (i.e., $\delta\mathbf{u}^{\pm,l}=\mathbf{u}^{\pm,l+\mathbf{1}}-\mathbf{u}^{\pm,l}$) is calculated using the linear system,

$$\mathbf{A}\delta\mathbf{u}^{\pm,l}=\mathbf{R}^l-\mathbf{A}\mathbf{u}^{\pm,l}.\tag{25}$$

This linear system makes it much easier to observe the well-behaved convergence as the right-hand side of the system is supposed to converge to identically 0. The linear system is solved using the PETSC Krylov subspace solver KSPBCGS which implements the BiCGStab (Stabilized version of BiConjugate Gradient) method with the relative convergence tolerance and absolute convergence tolerance equal to 10^{-16} and maximum iterations set to 100. In all our examples, the maximum iterations have not increased over 50.

3.2. Convergence test of the nonlinear solver with analytical surfaces

We study two characteristics: the order of accuracy of the numerical solution in the L^∞-norm, and the order of accuracy of the numerical gradients in the L^∞-norm.

Consider a spherical shell domain with inner and outer radii $r_i=0.151$ and $r_e=0.911$ and an immersed star-shaped interface described by the level-set function:

$$\phi_p=\sqrt{(x^2+y^2+z^2)}-r_o\left(1+\left(\frac{x^2+y^2}{x^2+y^2+z^2}\right)^2\sum_{k=1}^{3}\beta_k\cos\left(n_k\left(\arctan\left(\frac{y}{x}\right)-\theta_k\right)\right)\right),$$

with parameters:

$$r_o=0.483,\quad\begin{pmatrix}n_1\\ \beta_1\\ \theta_1\end{pmatrix}=\begin{pmatrix}3\\ 0.1\\ 0.5\end{pmatrix},\quad\begin{pmatrix}n_2\\ \beta_2\\ \theta_2\end{pmatrix}=\begin{pmatrix}4\\ -0.1\\ 1.8\end{pmatrix}\ \textit{and}\ \begin{pmatrix}n_3\\ \beta_3\\ \theta_3\end{pmatrix}=\begin{pmatrix}7\\ 0.15\\ 0\end{pmatrix}.\tag{26}$$

The problem geometry is illustrated in the Fig. 4.

Using the method of manufactured solutions, we analyze the following cases:

- Case 1: The exact solution is $u^-=\sin(2x)\cos(2y)\exp(z)$ and $u^+=\left(16\left(\frac{y-x}{3}\right)^5-20\left(\frac{y-x}{3}\right)^3+5\frac{y-x}{3}\right)\log(x+y+3)\cos(z)$. $\mu^-=10\,(1+(0.2\cos(x)+0.3\sin(y))\sin(z))$, $\mu^+=1$ and the nonlinear coefficients are set to $k^+=0.1$, $k^-=0$. The nonlinear function is $p^\pm=\sinh^\pm(u^\pm)$.
- Case 2: The exact solution is $u^-=\exp\left(x+z+y^2\right)(y+cos(x-z))$ and $u^+=\sin(x+0.3y)\cos(x-0.7y)\exp(z)+3\log\left(\sqrt{(x^2+y^2+z^2+0.5)}\right)$. $\mu^-=10\,(1+(0.2\cos(x)+0.3\sin(y))\sin(z))$, $\mu^+=1$ and the nonlinear coefficients are set to $k^+=0.1$, $k^-=0$. The nonlinear function is $p^\pm=\frac{1}{1+u^{\pm,2}}$.

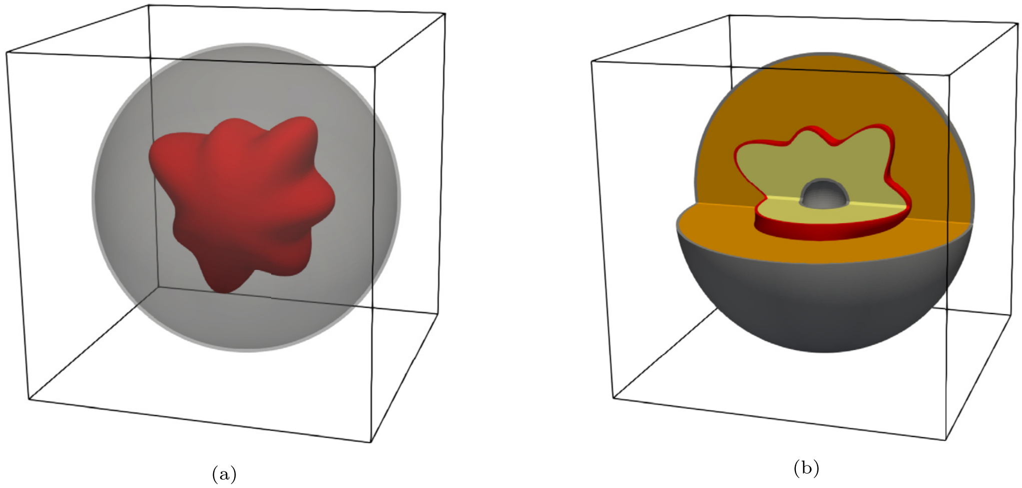


Fig. 4. Illustration of problem geometry in three-dimensional case.

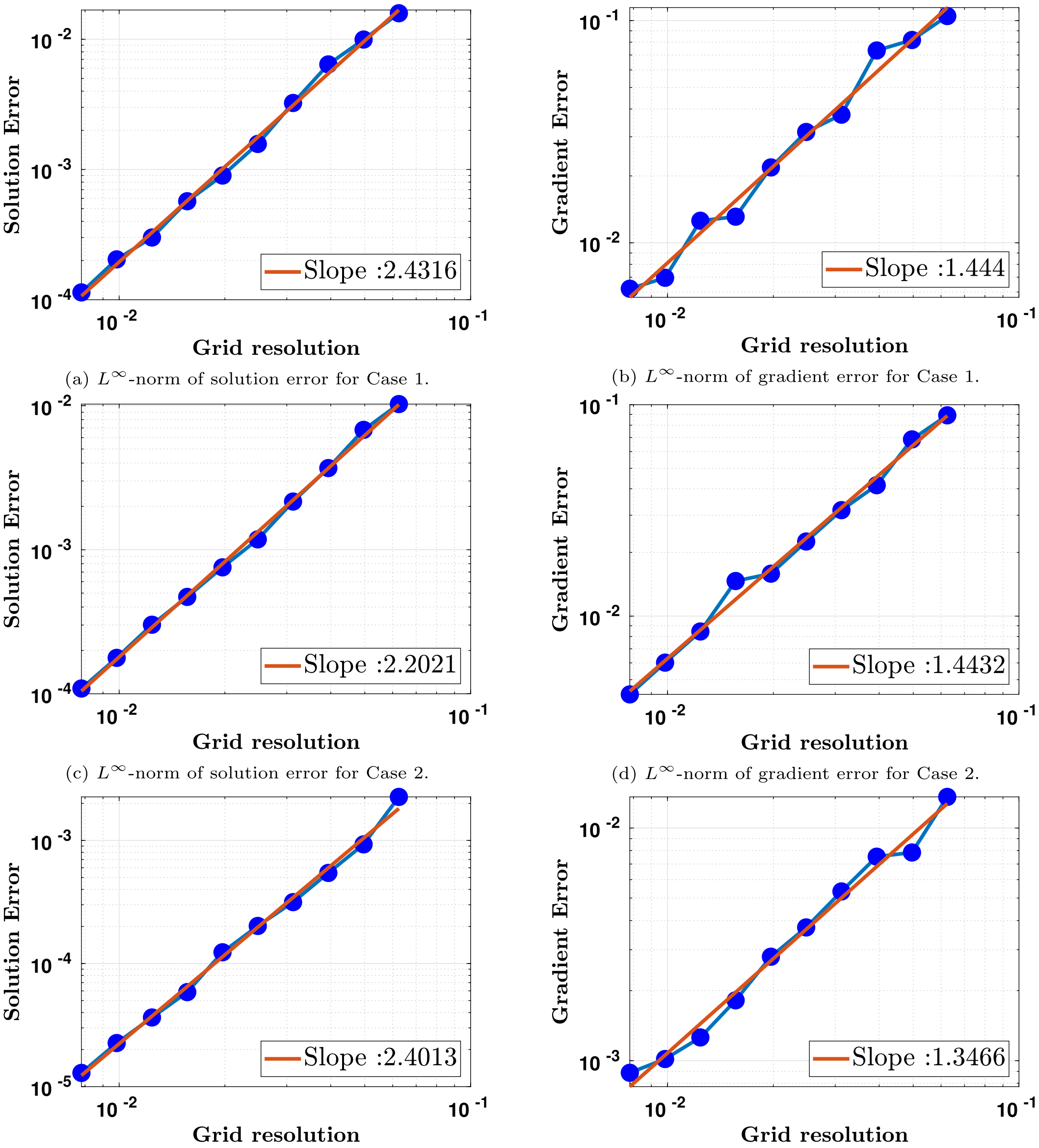
- Case 3: The exact solution is $u^- = 0.5\log\left(x + 0.5y - 0.3z + 3 + (x - 0.7y - 0.9z)^2\right)$ and $u^+ = \log\left(\frac{x+y+3}{y+z+3}\right)\sin(x + 0.5y + 0.7z)$. $\mu^- = 10\,(1 + (0.2\cos(x) + 0.3\sin(y))\sin(z))$, $\mu^+ = 1$ and the nonlinear coefficients are set to $k^+ = 0.1$, $k^- = 0$. The nonlinear function is $p^\pm = \sin(u^\pm)\cos(u^\pm)$.

The tests are carried out in uniform grids of different resolutions and the results are presented in Fig. 5. The nonlinear solver converges very quickly in few iterations as shown in Fig. 6. For convergence, the overall error in the domain is the maximum of the error in the positive and negative subdomains. The convergence test results indicate that the order of accuracy of numerical solutions is second-order (the linear fits of the log-log plots of cases 1, 2 and 3 have slopes of 2.4315, 2.2021 and 2.4013 respectively) with first-order gradients (the linear fits of the log-log plots of cases 1, 2 and 3 have slopes of 1.444, 1.4432 and 1.3466 respectively) in the L^∞-norm.

We test the convergence of the nonlinear solver on surface singularities as they might affect the order of accuracy of the solution as shown in [77]. The problem geometry chosen to analyze this is a sphere with the interface being an intersection of two spheres containing a surface singularity and is shown in Fig. 7a at the finest resolution $l_{\text{min}}/l_{\text{max}} = 8/10$, l_{min} and l_{max} refer to the minimum and maximum resolution of the grid. Using the method of manufactured solution, we analyze the case comprising the exact solution $u^- = 0.01\log\left(x + 0.5y - 0.3z + 3 + (x - 0.7y - 0.9z)^2\right)$ and $u^+ = 0.5\log\left(\frac{x+y+3}{y+z+3}\right)\sin(x + 0.5y + 0.7z)$. $\mu^- = 50$, $\mu^+ = 20$ and the nonlinear coefficients are set to $k^+ = k^- = 1$. The nonlinear function is $p^\pm = \sinh^\pm(u^\pm)$. The solution on a slice of the 3D domain is shown in Fig. 7b at the finest resolution $l_{\text{min}}/l_{\text{max}} = 8/10$. The presence of the surface singularity leads to a drop in the order of convergence of the solution from second order to first order shown in Fig. 7c and the convergence in gradient is lost.

3.3. Convergence test of the nonlinear solver with biomolecular surfaces as interfaces

The challenges involving surface singularities as illustrated in the last subsection become more difficult for proteins and so it is important to examine the convergence of the nonlinear solver with biomolecular surfaces. The `PDB` (Protein Data Bank) files of the biomolecules are converted to `PQR` (PQR files are PDB files where the occupancy and B-factor columns have been replaced by per-atom charge and radius) files using the `PDB2PQR` package [22,23] with the options `--drop-water` (this option drops water from the `PDB` file before processing protein) and `--ff-amber` (using the AMBER forcefield). The Solvent-Excluded Surface (SES) of the biomolecule is constructed using the method developed by Egan and Gibou [24]. Using the method of manufactured solution, we analyze the case comprising the exact solution $u^- = 0.01\log\left(x + 0.5y - 0.3z + 3 + (x - 0.7y - 0.9z)^2\right)$ and $u^+ = 0.5\log\left(\frac{x+y+3}{y+z+3}\right)\sin(x + 0.5y + 0.7z)$. $\mu^- = 50$, $\mu^+ = 20$ and the nonlinear coefficients are set to $k^+ = k^- = 1$. The nonlinear function is $p^\pm = \sinh^\pm(u^\pm)$. The domain is a cube with the SES of the biomolecule as an interface immersed inside it as illustrated in Fig. 8 for the biomolecule PDB:1ETN. Figs. 9 and 10 show how the solution error decreases in the negative and positive subdomain (inside and outside the biomolecule PDB:1ETN) respectively as the grid is refined from $l_{\text{min}}/l_{\text{max}} = 4/6$ to $l_{\text{min}}/l_{\text{max}} = 7/9$. The convergence plots for PDB:1ETN and another biomolecule PDB:1AJJ are shown in Fig. 11. Again the drop in convergence to 1st order in the order of accuracy of the numerical solution is due to the presence of surface singularities as seen in the previous example with the interface being an intersection of spheres. Since our solvation energy calculation (9) is dependent only on the electric potential which is the solution of the NPBE, the order of accuracy of the solvation energy will at least be first order.



(a) L^∞-norm of solution error for Case 1.

(b) L^∞-norm of gradient error for Case 1.

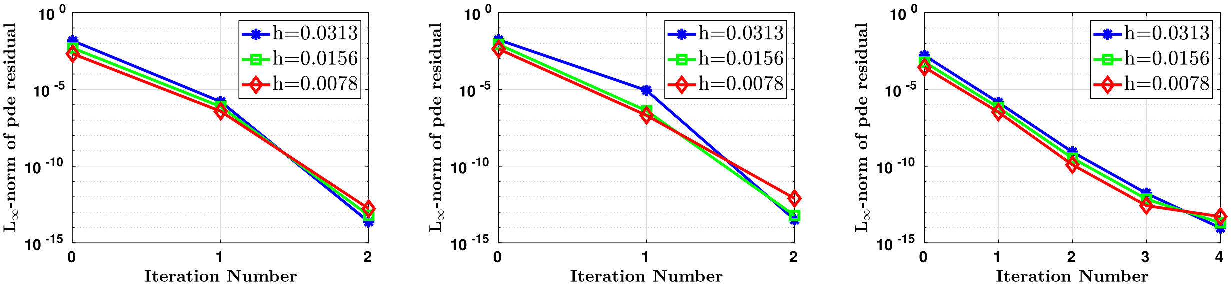
(c) L^∞-norm of solution error for Case 2.

(d) L^∞-norm of gradient error for Case 2.

(e) L^∞-norm of solution error for Case 3.

(f) L^∞-norm of gradient error for Case 3.

Fig. 5. Convergence tests in three spatial dimensions (each data point represents maximum value among $5 \times 5 \times 5 = 125$ different relative placements of an immersed interface in the computational grid).

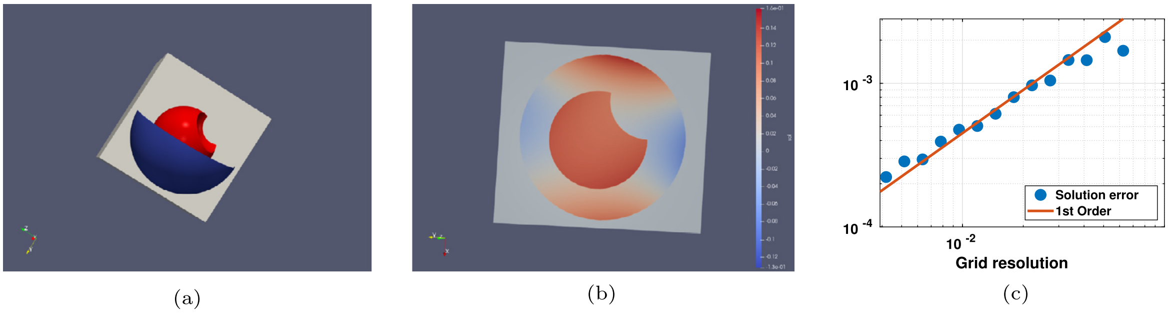


(a)

(b)

(c)

Fig. 6. L^∞-norm of PDE Residual, as referred to in (25), vs Iteration Number demonstrating how quickly the nonlinear solver converges for different grid resolutions(h) for Case 1 (a), Case 2 (b) and Case 3 (c).



(a) (b) (c)

Fig. 7. a) Illustration of problem geometry b) Solution of the nonlinear problem with surface singularity c) L^∞-norm of the solution error.

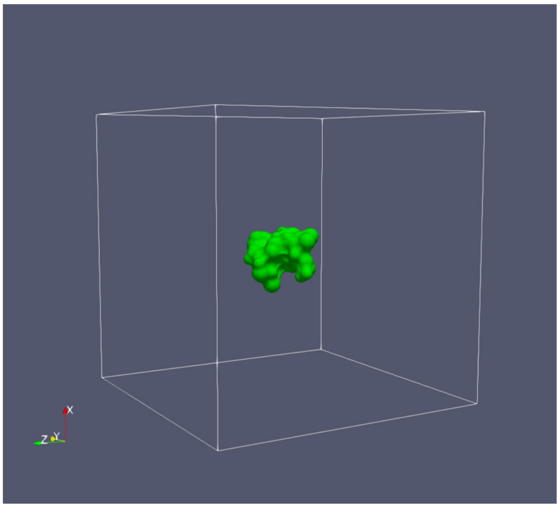
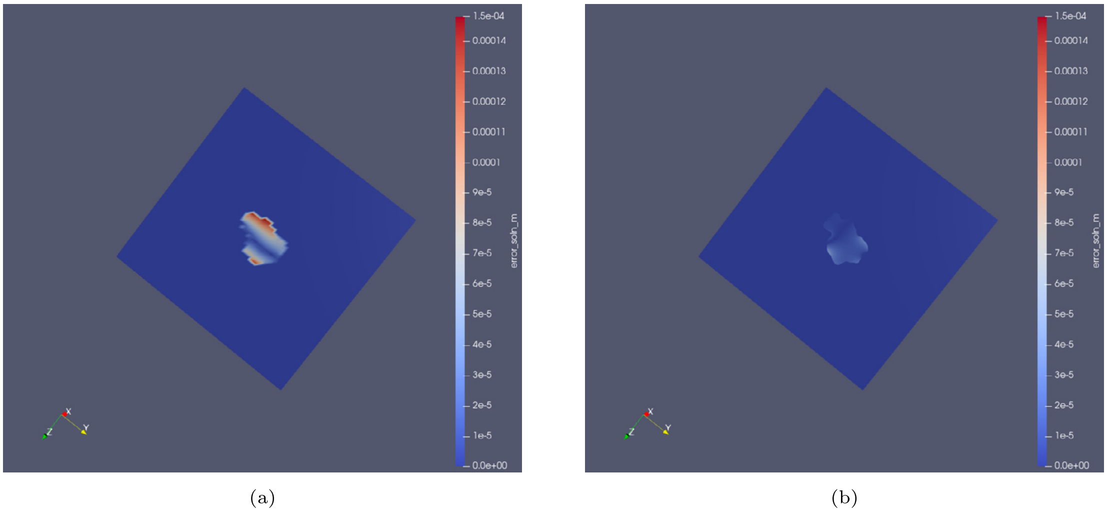
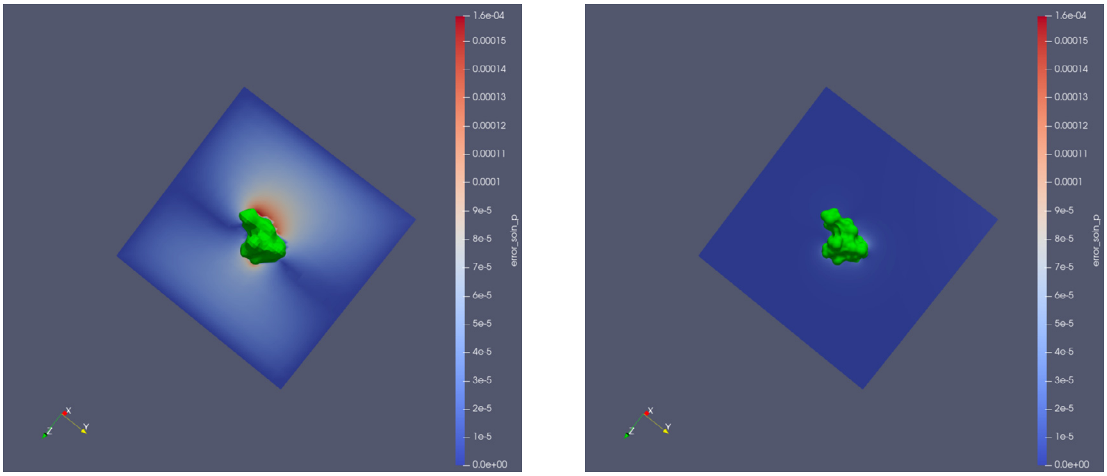


Fig. 8. Illustration of domain with biomolecular surface of PDB:1ETN as interface.



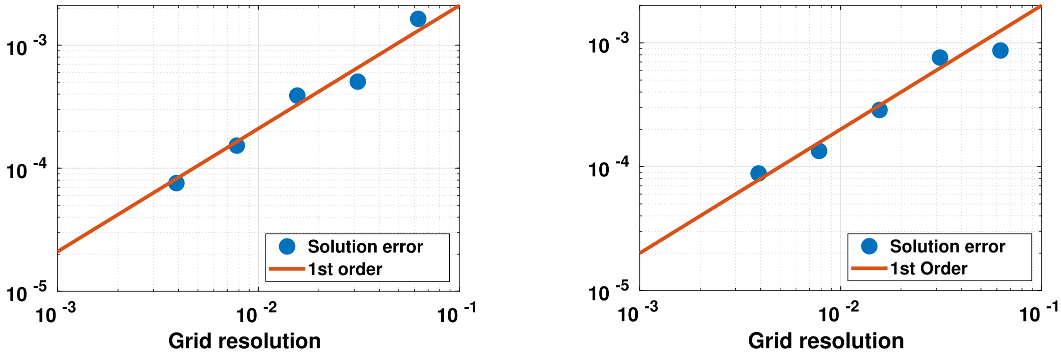
(a) (b)

Fig. 9. L^∞-norm of the error in the negative subdomain a) $l_{\min}/l_{\max} = 4/6$ b) $l_{\min}/l_{\max} = 7/9$.



(a) (b)

Fig. 10. L^∞-norm of the error in the positive subdomain a) $l_{\min}/l_{\max} = 4/6$ b) $l_{\min}/l_{\max} = 7/9$.



(a) (b)

Fig. 11. Convergence test in 3D with biomolecular interfaces: a) `1etn` b) `1ajj`.

4. Electrostatics and solvation energy of biomolecules

4.1. Kirkwood's solution

To test the solver, we start from the Kirkwood model, which has an explicit analytical solution (see [43,27,61]). In this model, there is a single sphere of radius 2 Å with the origin at its center, which contains multiple point-distributed charges. The dielectric permittivity in the solute cavity, as in vacuum, is $\varepsilon_{\mathrm{r,m}} = 1$ and the solvent is water with the dielectric permittivity $\varepsilon_{\mathrm{r,s}} = 78.54$ at the room temperature $T = 298.15$ K. The nonlinear factor $\kappa = 0$, which makes the problem linear and hence we solve the simplified problem:

$$\begin{cases} -\nabla\cdot\left(\varepsilon_{\mathrm{r}}\nabla\psi\right) = \sum_k \frac{q_k\tilde{e}^2 z}{\tilde{\varepsilon}_\circ\tilde{k}_{\mathrm{B}}\tilde{T}\tilde{\ell}}\delta\left(\mathbf{x}-\mathbf{x}_k\right), & \\ \psi = \sum_k \left(\frac{q_k\tilde{e}^2 z}{\tilde{\varepsilon}_\circ\tilde{k}_{\mathrm{B}}\tilde{T}\tilde{\ell}}\right)\frac{1}{4\pi\,\varepsilon_{\mathrm{r,s}}\|\mathbf{x}-\mathbf{x}_k\|} & \text{on } \partial\Omega, \\ [\psi] = 0 & \text{across } \Gamma, \\ [\varepsilon_{\mathrm{r}}\nabla\psi\cdot\mathbf{n}] = 0 & \text{across } \Gamma, \end{cases} \tag{27}$$

and the free solvation energy $\mathcal{E}^{\mathrm{s}}$ is

$$\mathcal{E}^{\mathrm{s}} = \tilde{k}_{\mathrm{B}}\tilde{T}\sum_k \frac{q_k}{2z}\check{\psi}\left(\mathbf{x}_k\right). \tag{28}$$

The free solvation energy obtained from the solver is compared to the value obtained from the analytical solution given in equation (13) of [43] in the following cases:

Table 1
Electrostatic solvation energies ($\mathcal{E}^s$) for different Kirkwood models. l_{min} and l_{max} refer to the minimum and maximum resolutions of the grid. The refinement is done based on the distance from the surface of the sphere, with the finest grid closest to the surface. R.E. represents the relative error with respect to the exact solution.

Case 0: $\mathcal{E}^s_{\text{ex.}} = -81.9589$ kcal mol^{-1}				
l_{min}	l_{max}	h (Å)	$\mathcal{E}^s$ (kcal mol^{-1})	R.E.
5	7	0.1804	−81.9582	8.5409$e-6$
6	8	0.0902	−81.9587	1.2201$e-6$
7	9	0.0451	−81.9589	0
Case 1: $\mathcal{E}^s_{\text{ex.}} = -349.5051$ kcal mol^{-1}				
l_{min}	l_{max}	h (Å)	$\mathcal{E}^s$ (kcal mol^{-1})	R.E.
5	7	0.1804	−349.628	3.5736$e-4$
6	8	0.0902	−349.566	1.7425$e-4$
7	9	0.0451	−349.53	7.4105$e-5$
Case 2: $\mathcal{E}^s_{\text{ex.}} = -62.7523$ kcal mol^{-1}				
l_{min}	l_{max}	h (Å)	$\mathcal{E}^s$ (kcal mol^{-1})	R.E.
5	7	0.1804	−62.948	3.1$e-3$
6	8	0.0902	−62.8155	1.0$e-3$
7	9	0.0451	−62.7777	4.0477$e-4$
Case 3: $\mathcal{E}^s_{\text{ex.}} = -2988.5210$ kcal mol^{-1}				
l_{min}	l_{max}	h (Å)	$\mathcal{E}^s$ (kcal mol^{-1})	R.E.
5	7	0.1804	−2989.09	1.904$e-4$
6	8	0.0902	−2988.88	1.2013$e-4$
7	9	0.0451	−2988.69	5.655$e-5$

- Case 0. (Born model): One positive unit charge placed at $(0, 0, 0)$.
- Case 1. Two positive unit charges placed at $(1, 0, 0)$ and $(-1, 0, 0)$.
- Case 2. Two positive unit charges placed at $(1, 0, 0)$ and $(-1, 0, 0)$, and two negative unit charges placed at $(0, 1, 0)$ and $(0, -1, 0)$.
- Case 3. Six positive unit charges placed at $(0.4, 0, 0)$, $(0, 0.8, 0)$, $(0, 0, 1.2)$, $(0, 0, -0.4)$, $(-0.8, 0, 0)$ and $(0, -1.2, 0)$.

Table 1 lists the electrostatic solvation energy obtained from our solver, as well as the relative errors (R.E.), which are computed with the following definition:

$$\text{R.E.} = \frac{\left|\mathcal{E}^s_{\text{ex.}} - \mathcal{E}^s\right|}{\left|\mathcal{E}^s_{\text{ex.}}\right|}. \tag{29}$$

From Table 1, it is observed that as the grid is refined, R.E. decreases, which is in line with the convergence of our solver.

4.2. Biomolecules - comparison with APBS

In this section, the electrostatic solvation energies calculated by our solver are compared to the ones from APBS for five biomolecules. APBS provides several choices of surface definitions; we use `srfm = mol`. In this definition, the dielectric coefficient is based on a molecular surface definition. The problem domain is split into two parts: the "free volume" space (V_1) is defined by the collection of solvent-sized spheres (the radius of which are specified by the keyword `srad` and this volume does not overlap with the biomolecular atoms) and its complement (V_2) containing all the biomolecular atoms. The ion-accessibility coefficient is defined using the "inflated" Van der Waals model. This model also involves splitting the domain into two parts. The first part (V_1'), which is formed by the union of biomolecular atoms and their radius is increased by the radius of the ion species (specified by `ion`) and the complement of it is V_2'. For the SES surface definition used in our solver, V_1 is the same as V_2' and to have a one to one comparison with APBS, both the radius of ion species (with the keyword `ion`) and `srad` are set to 1.4 Å. The multi-grid solver in APBS is used by setting `mg-manual`. The other parameters are set to `gcent = mol 1`, `bcfl = mdh`, `chgm = spl4`, `sdens = 10`, and `swin = 0.3`. The dielectric permittivity in the solute cavity, as in vacuum, is $\varepsilon_{\text{r,m}} = 1$ and the solvent is water with the dielectric permittivity $\varepsilon_{\text{r,s}} = 78.54$ at the room temperature $T = 298.15$ K. The modified Debye length $\tilde{\lambda}$ is 1.08575 Å for an ionic strength of 0.1 molar. In our solver, we use zero boundary condition instead of multiple Debye-Hückel, which is used in APBS, and to ensure that this boundary condition produces accurate results, the biomolecule is scaled such that it covers 30% of the domain and is placed at the center of the domain. Table 2 gives the solvation free energies from APBS and our solver for five different biomolecules (APBS has a uniform grid whereas our solver has an adaptive grid with the finest grid cells closest to the surface of the biomolecule). The comparison shows the close agreement between the solvation energies obtained from both the solvers for different biomolecules.

Table 2
Electrostatic solvation energies ($\mathcal{E}^s$) for five biomolecules using APBS (a) and our solver (b) are compared. h_g represent the grid spacing in APBS. For our solver, the biomolecule is scaled in a way that it covers only 30% of the domain and is positioned at the center of the domain. $l_{\min}$ and $l_{\max}$ refer to the minimum and maximum resolution of the grid and the length of the finest cell-diagonal is h. The refinement is done based on the distance from the surface of the biomolecule, with the finest grid cells closest to the surface.

$\mathcal{E}^s$ using APBS		
Molecule	h_g (Å)	$\mathcal{E}^s$ (kcal mol^{-1})
1ETN	0.214	−235.6279
1ETN	0.107	−234.0234
1AJJ	0.214	−1105.1052
1AJJ	0.107	−1097.1249
1BBL	0.219	−977.6459
1BBL	0.109	−967.6241
1BOR	0.219	−863.2013
1BOR	0.109	−853.6672
1PTQ	0.214	−826.1387
1PTQ	0.107	−814.9547

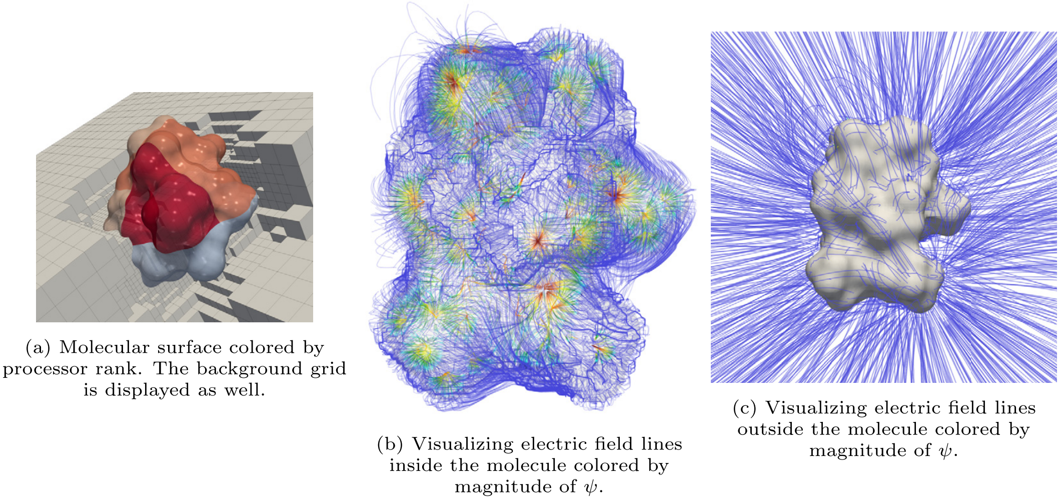
(a) APBS

$\mathcal{E}^s$ using our Solver				
Molecule	$l_{\min}$	$l_{\max}$	h (Å)	$\mathcal{E}^s$ (kcal mol^{-1})
1ETN	7	9	0.217	−232.5507
1ETN	8	10	0.108	−233.2809
1AJJ	7	9	0.217	−1088.6431
1AJJ	8	10	0.108	−1095.1261
1BBL	7	9	0.217	−958.1678
1BBL	8	10	0.108	−962.0968
1BOR	7	9	0.217	−840.4489
1BOR	8	10	0.108	−846.8078
1PTQ	7	9	0.217	−803.8852
1PTQ	8	10	0.108	−808.6974

(b) Our Solver

Table 3
Electrostatic solvation energies ($\mathcal{E}^s$) for PDB:1ETN calculated using our solver. $l_{\min}$ and $l_{\max}$ refer to the minimum and maximum resolution of the grid, h is the length of the finest cell diagonal and box size refers to the fraction, along one Cartesian dimension only, of the domain covered by the molecule. The refinement is done based on the distance from the biomolecule's surface, with the finest grid cells closest to the surface.

$\mathcal{E}^s$ for 1etn				
$l_{\min}$	$l_{\max}$	h (Å)	box size	$\mathcal{E}^s$ (kJ/mol)
8	10	0.1149	0.3	−233.2809
8	11	0.05578	0.15	−233.2407
8	12	0.02789	0.075	−233.2682


(a) Molecular surface colored by processor rank. The background grid is displayed as well.

(b) Visualizing electric field lines inside the molecule colored by magnitude of ψ.

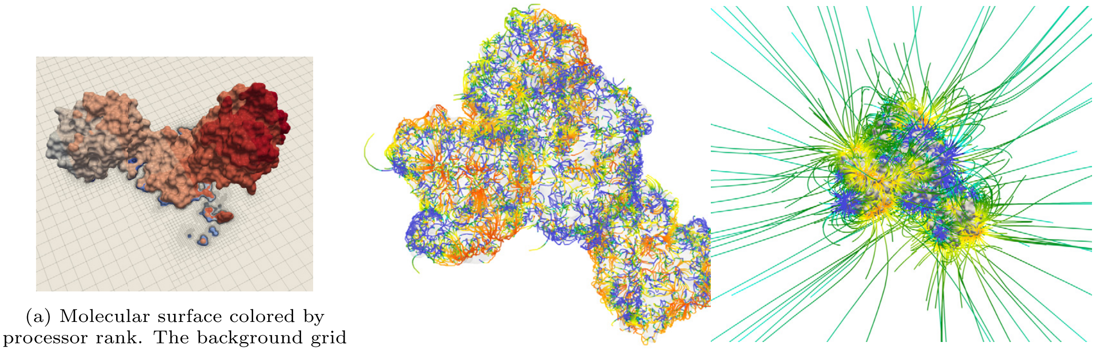
(c) Visualizing electric field lines outside the molecule colored by magnitude of ψ.

Fig. 12. Example of solution obtained by the current solver with the PDB:1ETN molecule.

In Table 3, the biomolecule is scaled to varying box sizes from 30% to 7.5% and the solvation free energy values obtained from the solver illustrate the benefits of the sharp treatment of the biomolecular interface and the adaptive grid in our framework enabling us to confine the biomolecule in such a small portion of the domain and still resolve the surface enough to produce accurate solvation free energy values.

4.3. Strong scalability

In this section, we show the strong scalability of our solver by focusing on three biomolecules of increasing size - PDB:1ETN (82 atoms), PDB:1A2K (13627 atoms) and PDB:3J6D (131664 atoms). The simulations were run on Stampede2 cluster at the Texas Advanced Computing Center on SKX nodes - each of which have 48 cores on two sockets, 2.1 GHz

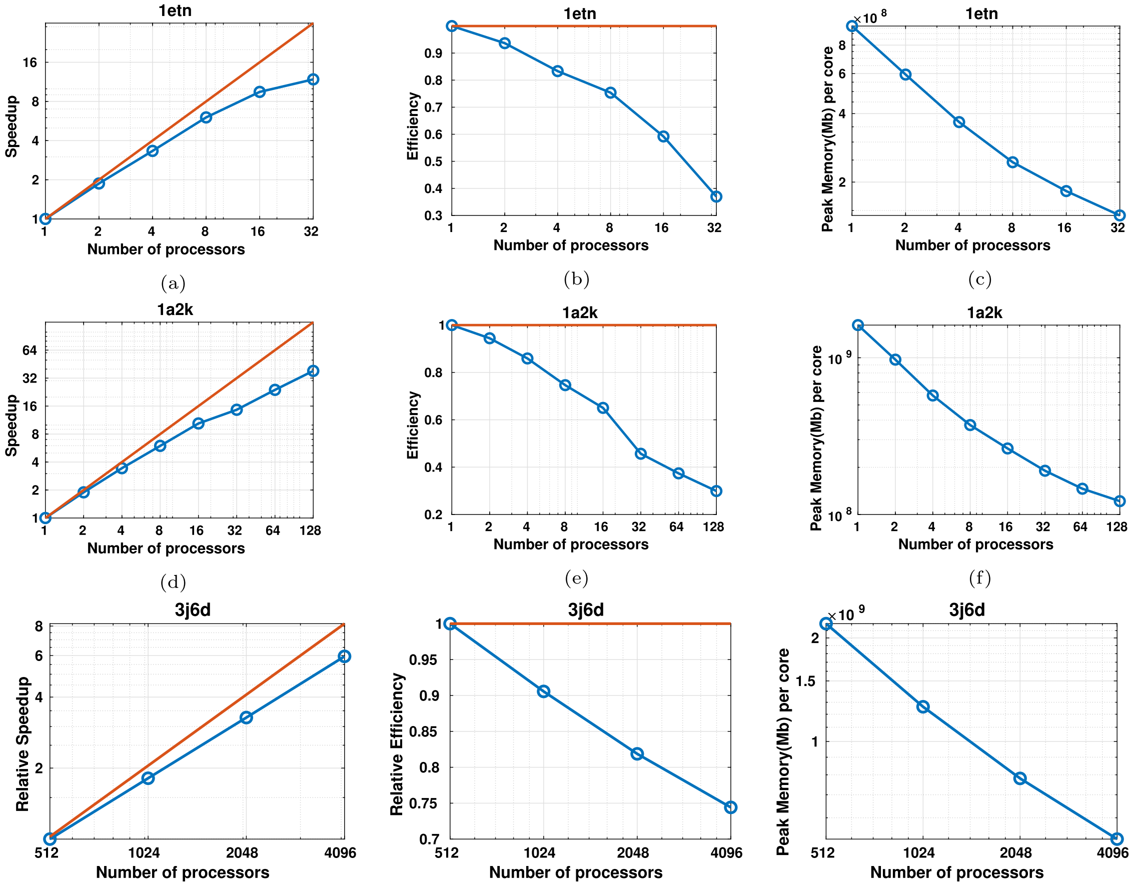


(a) Molecular surface colored by processor rank. The background grid is displayed as well.

(b) Visualizing electric field lines inside the molecule colored by magnitude of ψ.

(c) Visualizing electric field lines outside the molecule colored by magnitude of ψ.

Fig. 13. Example of solution obtained by the current solver with PDB:1A2K molecule.



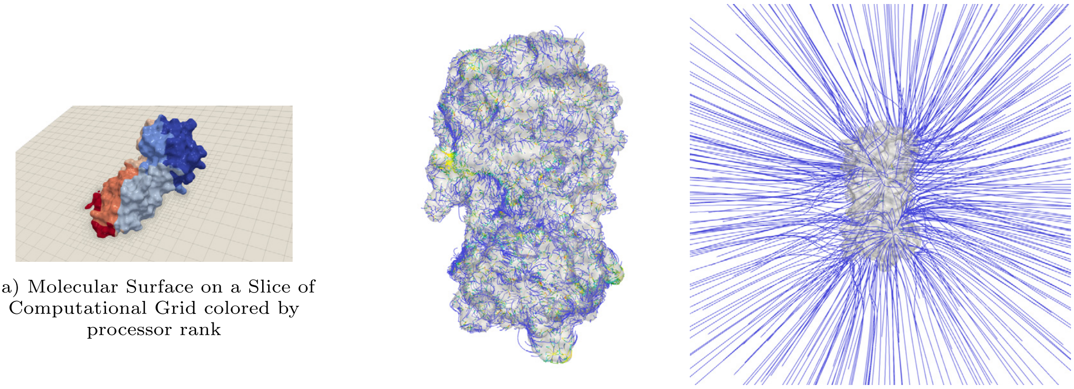
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 14. Strong Scalability of the solver with three molecules- PDB:1ETN (82 atoms), PDB:1A2K (13627 atoms), and PDB:3J6D (131664 atoms) and the respective Peak Memory Usage per core. For PDB:1ETN and PDB:1A2K, it was possible to run the code in a single core and hence the speedup with respect to a single core could be calculated. For PDB:3J6D, a much finer grid is needed and it was not possible to run it in a single core and hence a relative speedup with respect to the run in 512 cores is shown.

nominal (1.4 - 3.7 GHz depending on instruction set and number of active cores), 192 GB (2.67 GHz) DDR4 RAM and each socket can cache up to 57 MB. Figs. 12 and 13 show the molecular surface in the adaptive grid and the electric field lines inside and outside the biomolecule for PDB:1ETN and PDB:1A2K respectively. Fig. 14 shows the strong scalability of our solver for all the three molecules.



(a) Molecular Surface on a Slice of Computational Grid colored by processor rank

(b) Visualizing electric field lines interior of the biomolecule colored by magnitude of ψ.

(c) Visualizing electric field lines exterior of the biomolecule.

Fig. 15. Crystal structure of the COVID-19 (PDB:6LU7).

4.4. *Electrostatics of 6lu7*

PDB:6LU7 is the crystal structure of the COVID-19 main protease in complex with an inhibitor N3 (Fig. 15). Given the relevance to the pandemic of 2020, we resolve the electrostatics around the biomolecule with the grid parameters being $l_{\text{min}} = 8$, $l_{\text{max}} = 10$ and $h = 0.11149$ Å. Its solvation energy is -11324.8 kJ/mol.

5. Conclusion

We have incorporated the idea of a sharp treatment of jump conditions as a central framework for the development of a scalable and accurate Nonlinear Poisson-Boltzmann solver for biomolecular electrostatic computation on Octree adaptive Cartesian grids in parallel. This solver, in association with the fast surface construction developed in [24], offers new capabilities in a distributed computing framework for calculating electric field of large biomolecules. The capabilities of the code have been illustrated by the computation of electric field and electrostatic solvation energies and by comparing the results with the APBS solver. The solver's scalability on over four thousand cores was also demonstrated.

CRediT authorship contribution statement

Rochishnu Chowdhury, Raphael Egan and Professor Frederic Gibou conceived of the idea. Daniil Bochkov helped Rochishnu Chowdhury and Raphael Egan to develop the nonlinear elliptic equation solver under the supervision of Professor Frederic Gibou. Biomolecular electrostatic calculations were done by Rochishnu Chowdhury. The first draft of the manuscript was written by Rochishnu Chowdhury with edits done by the rest of the authors.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We wish to thank Dr. Nathan Baker for very helpful discussions on this topic. The research was partially funded by the Army Research Office under grant number ARO W911NF-16-1-0136.

References

[1] L. Adams, T.P. Chartier, New geometric immersed interface multigrid solvers, SIAM J. Sci. Comput. 25 (5) (2004) 1516–1533.

[2] L. Adams, T.P. Chartier, A comparison of algebraic multigrid and geometric immersed interface multigrid methods for interface problems, SIAM J. Sci. Comput. 26 (3) (2005) 762–784.

[3] N. Baker, M. Holst, F. Wang, Adaptive multilevel finite element solution of the Poisson–Boltzmann equation II. Refinement at solvent-accessible surfaces in biomolecular systems, J. Comput. Chem. 21 (15) (2000) 1343–1352.

[4] N.A. Baker, D. Sept, S. Joseph, M.J. Holst, J.A. McCammon, Electrostatics of nanosystems: application to microtubules and the ribosome, Proc. Natl. Acad. Sci. 98 (18) (2001) 10037–10041.

[5] J.P. Bardhan, M.D. Altman, D.J. Willis, S.M. Lippow, B. Tidor, J.K. White, Numerical integration techniques for curved-element discretizations of molecule-solvent interfaces, J. Chem. Phys. 127 (1) (2007) 014701.

[6] D. Bashford, An object-oriented programming suite for electrostatic effects in biological molecules: an experience report on the mead project, in: International Conference on Computing in Object-Oriented Parallel Environments, Springer, 1997, pp. 233–240.
[7] P.A. Berthelsen, A decomposed immersed interface method for variable coefficient elliptic equations with non-smooth and discontinuous solutions, J. Comput. Phys. 197 (1) (2004) 364–386.
[8] D. Bochkov, F. Gibou, Solving Poisson-type equations with Robin boundary conditions on piecewise smooth interfaces, J. Comput. Phys. 376 (2019) 1156–1198.
[9] D. Bochkov, F. Gibou, Solving elliptic interface problems with jump conditions on Cartesian grids, J. Comput. Phys. (2020) 109269.
[10] S.D. Bond, J.H. Chaudhry, E.C. Cyr, L.N. Olson, A first-order system least-squares finite element method for the Poisson-Boltzmann equation, J. Comput. Chem. 31 (8) (2010) 1625–1635.
[11] A.H. Boschitsch, M.O. Fenley, Hybrid boundary element and finite difference method for solving the nonlinear Poisson–Boltzmann equation, J. Comput. Chem. (ISSN 1096-987X) 25 (7) (2004) 935–955, https://doi.org/10.1002/jcc.20000.
[12] 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.
[13] Q. Cai, M.-J. Hsieh, J. Wang, R. Luo, Performance of nonlinear finite-difference Poisson- Boltzmann solvers, J. Chem. Theory Comput. 6 (1) (2010) 203–211.
[14] D.L. Chapman, LI. A contribution to the theory of electrocapillarity, Philos. Mag. 25 (148) (1913) 475–481.
[15] J.H. Chaudhry, S.D. Bond, L.N. Olson, A weighted adaptive least-squares finite element method for the Poisson–Boltzmann equation, Appl. Comput. Math. 218 (9) (2012) 4892–4902.
[16] D. Chen, Z. Chen, C. Chen, W. Geng, G.-W. Wei MIBPB, A software package for electrostatic analysis, J. Comput. Chem. (ISSN 1096-987X) 32 (4) (2011) 756–770, https://doi.org/10.1002/jcc.21646.
[17] L. Chen, M.J. Holst, J. Xu, The finite element approximation of the nonlinear Poisson–Boltzmann equation, SIAM J. Numer. Anal. 45 (6) (2007) 2298–2320.
[18] T. Chen, J. Strain, Piecewise-polynomial discretization and Krylov-accelerated multigrid for elliptic interface problems, J. Comput. Phys. 227 (16) (2008) 7503–7542.
[19] M.E. Davis, J.A. McCammon, Electrostatics in biomolecular structure and dynamics, Chem. Rev. 90 (3) (1990) 509–521.
[20] P. Debye, E. Hückel, The theory of electrolytes. I. Freezing point depres-sion and related phenomena (Zur theorie der elektrolyte. I. Gefrierpunkt-serniedrigung und verwandte erscheinungen), Phys. Z. 24 (1923) 185–206.
[21] B.v. Derjaguin, L. Landau, Theory of the stability of strongly charged lyophobic sols and of the adhesion of strongly charged particles in solutions of electrolytes, Prog. Surf. Sci. 43 (1–4) (1993) 30–59.
[22] 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.
[23] T.J. Dolinsky, P. Czodrowski, H. Li, J.E. Nielsen, J.H. Jensen, G. Klebe, N.A. Baker, Pdb2pqr: expanding and upgrading automated preparation of biomolecular structures for molecular simulations, Nucleic Acids Res. 35 (suppl_2) (2007) W522–W525.
[24] R. Egan, F. Gibou, Fast and scalable algorithms for constructing solvent-excluded surfaces of large biomolecules, J. Comput. Phys. 374 (2018) 91–120.
[25] R. Egan, F. Gibou xgfm, Recovering convergence of fluxes in the ghost fluid method, J. Comput. Phys. (2020) 109351.
[26] R.P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. (ISSN 0021-9991) 152 (2) (1999) 457–492, https://doi.org/10.1006/jcph.1999.6236, http://www.sciencedirect.com/science/article/pii/S0021999199962368.
[27] W. Geng, A boundary integral Poisson-Boltzmann solvers package for solvated biomolecular simulations, Comput. Math. Biophys. 3 (1) (2015).
[28] W. Geng, S. Zhao, A two-component matched interface and boundary (mib) regularization for charge singularity in implicit solvation, J. Comput. Phys. 351 (2017) 25–39.
[29] F. Gibou, R. Fedkiw, A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem, J. Comput. Phys. (ISSN 0021-9991) 202 (2) (2005) 577–601, https://doi.org/10.1016/j.jcp.2004.07.018, http://www.sciencedirect.com/science/article/pii/S0021999104002980.
[30] F. Gibou, R.P. Fedkiw, L.-T. Cheng, M. Kang, A second-order-accurate symmetric discretization of the Poisson equation on irregular domains, J. Comput. Phys. (ISSN 0021-9991) 176 (1) (2002) 205–227, https://doi.org/10.1006/jcph.2001.6977, http://www.sciencedirect.com/science/article/pii/S0021999101969773.
[31] F. Gibou, R. Fedkiw, S. Osher, A review of level-set methods and some recent applications, J. Comput. Phys. (ISSN 0021-9991) 353 (Supplement C) (2018) 82–109, https://doi.org/10.1016/j.jcp.2017.10.006, http://www.sciencedirect.com/science/article/pii/S0021999117307441.
[32] M.K. Gilson, B. Honig, Calculation of the total electrostatic energy of a macromolecular system: solvation energies, binding energies, and conformational analysis, Proteins, Struct. Funct. Bioinform. 4 (1) (1988) 7–18.
[33] A. Guittet, M. Lepilliez, S. Tanguy, F. Gibou, Solving elliptic problems with discontinuities on irregular domains – the Voronoi interface method, J. Comput. Phys. 322 (2015) 345–364.
[34] A. Guittet, C. Poignard, F. Gibou, A Voronoi interface approach to cell aggregate electropermeabilization, J. Comput. Phys. (ISSN 0021-9991) 332 (2017) 143–159, https://doi.org/10.1016/j.jcp.2016.11.048, http://www.sciencedirect.com/science/article/pii/S0021999116306404.
[35] G. Guoy, Constitution of the electric charge at the surface of an electrolyte, J. Phys. 9 (1910) 457–467.
[36] A. Helgadóttir, F. Gibou, A Poisson–Boltzmann solver on irregular domains with Neumann or Robin boundary conditions on non-graded adaptive grid, J. Comput. Phys. 230 (10) (May 2011) 3830–3848.
[37] M. Holst, Adaptive numerical treatment of elliptic systems on manifolds, Adv. Comput. Math. 15 (1–4) (2001) 139–191.
[38] M. Holst, N. Baker, F. Wang, Adaptive multilevel finite element solution of the Poisson–Boltzmann equation I. Algorithms and examples, J. Comput. Chem. 21 (15) (2000) 1319–1342.
[39] B. Honig, A. Nicholls, Classical electrostatics in biology and chemistry, Science 268 (5214) (1995) 1144–1149.
[40] S. Jo, M. Vargyas, J. Vasko-Szedlar, B. Roux, W. Im, Pbeq-solver for online visualization of electrostatic potential of biomolecules, Nucleic Acids Res. (suppl_2) (2008) W270–W275.
[41] A. Juffer, E.F. Botta, B.A. van Keulen, A. van der Ploeg, H.J. Berendsen, The electric potential of a macromolecule in a solvent: a fundamental approach, J. Comput. Phys. 97 (1) (1991) 144–171.
[42] E. Jurrus, D. Engel, K. Star, K. Monson, J. Brandi, L.E. Felberg, D.H. Brookes, L. Wilson, J. Chen, K. Liles, et al., Improvements to the apbs biomolecular solvation software suite, Protein Sci. 27 (1) (2018) 112–128.
[43] J.G. Kirkwood, Theory of solutions of molecules containing widely separated charges with special application to zwitterions, J. Chem. Phys. 2 (7) (1934) 351–361.
[44] R. LeVeque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (1994) 1019–1044.
[45] L. Li, C. Li, S. Sarkar, J. Zhang, S. Witham, Z. Zhang, L. Wang, N. Smith, M. Petukh, E. Alexov, Delphi: a comprehensive suite for delphi software and associated resources, BMC Biophys. 5 (1) (2012) 9.

[46] Z. Li, A fast iterative algorithm for elliptic interface problems, SIAM J. Numer. Anal. 35 (1998) 230–254.

[47] Z. Li, K. Ito, The Immersed Interface Method – Numerical Solutions of PDEs Involving Interfaces and Irregular Domains, SIAM Frontiers in Applied Mathematics, vol. 33, 2006.

[48] J. Liang, S. Subramaniam, Computation of molecular electrostatics with boundary element methods, Biophys. J. 73 (4) (1997) 1830–1841.

[49] F. Lipparini, B. Stamm, E. Cances, Y. Maday, B. Mennucci, Fast domain decomposition algorithm for continuum solvation models: energy and first derivatives, J. Chem. Theory Comput. 9 (8) (2013) 3637–3648.

[50] F. Lipparini, L. Lagardère, G. Scalmani, B. Stamm, E. Cancès, Y. Maday, J.-P. Piquemal, M.J. Frisch, B. Mennucci, Quantum calculations in solution for large to very large molecules: a new linear scaling qm/continuum approach, J. Phys. Chem. Lett. 5 (6) (2014) 953–958.

[51] F. Lipparini, G. Scalmani, L. Lagardère, B. Stamm, E. Cancès, Y. Maday, J.-P. Piquemal, M.J. Frisch, B. Mennucci, Quantum, classical, and hybrid qm/mm calculations in solution: general implementation of the ddcosmo linear scaling strategy, J. Chem. Phys. 141 (18) (2014) 184108.

[52] X.-D. Liu, R.P. Fedkiw, M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, J. Comput. Phys. (ISSN 0021-9991) 160 (1) (2000) 151–178, https://doi.org/10.1006/jcph.2000.6444, http://www.sciencedirect.com/science/article/pii/S0021999100964441.

[53] J.D. Madura, J.M. Briggs, R.C. Wade, M.E. Davis, B.A. Luty, A. Ilin, J. Antosiewicz, M.K. Gilson, B. Bagheri, L.R. Scott, et al., Electrostatics and diffusion of molecules in solution: simulations with the university of Houston Brownian dynamics program, Comput. Phys. Commun. 91 (1–3) (1995) 57–95.

[54] A.M. Micu, B. Bagheri, A.V. Ilin, R. Scott, B.M. Pettitt, Numerical considerations in the computation of the electrostatic free energy of interaction within the Poisson–Boltzmann theory, J. Comput. Phys. 136 (2) (1997) 263–271.

[55] C. Min, F. Gibou, Geometric integration over irregular domains with application to level set methods, J. Comput. Phys. 226 (2007) 1432–1443.

[56] C. Min, F. Gibou, H. Ceniceros, A supra-convergent finite difference scheme for the variable coefficient Poisson equation on non-graded grids, J. Comput. Phys. 218 (2006) 123–140.

[57] M. Mirzadeh, M. Theillard, F. Gibou, A second-order discretization of the nonlinear Poisson-Boltzmann equation over irregular geometries using non-graded adaptive Cartesian grids, J. Comput. Phys. 230 (5) (Dec. 2010) 2125–2140.

[58] M. Mirzadeh, M. Theillard, A. Helgadottir, D. Boy, F. Gibou, An adaptive, finite difference solver for the nonlinear Poisson–Boltzmann equation with applications to biomolecular computations, Commun. Comput. Phys. 13 (1) (2012) 150–173.

[59] P. Mistani, A. Guittet, C. Poignard, F. Gibou, A parallel Voronoi-based approach for mesoscale simulations of cell aggregate electropermeabilization, J. Comput. Phys. 380 (2019) 48–64.

[60] Y.T. Ng, C. Min, F. Gibou, An efficient fluid–solid coupling algorithm for single-phase flows, J. Comput. Phys. 228 (23) (Dec. 2009) 8807–8829.

[61] D.D. Nguyen, B. Wang, G.-W. Wei Accurate, Robust, and reliable calculations of Poisson–Boltzmann binding energies, J. Comput. Chem. 38 (13) (2017) 941–948.

[62] S. Osher, J. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.

[63] J. Papac, F. Gibou, C. Ratsch, Efficient symmetric discretization for the Poisson, heat and Stefan-type problems with Robin boundary conditions, J. Comput. Phys. 229 (3) (Feb. 2010) 875–889.

[64] N.V. Prabhu, M. Panda, Q. Yang, K.A. Sharp, Explicit ion, implicit water solvation for molecular dynamics of nucleic acids and highly charged molecules, J. Comput. Chem. 29 (7) (2008) 1113–1130.

[65] L.R. Pratt, G.J. Tawa, G. Hummer, A.E. García, S.A. Corcelli, Boundary integral methods for the Poisson equation of continuum dielectric solvation models, Int. J. Quant. Chem. 64 (1) (1997) 121–141.

[66] C. Quan, B. Stamm, Y. Maday, A domain decomposition method for the Poisson–Boltzmann solvation models, SIAM J. Sci. Comput. 41 (2) (2019) B320–B350.

[67] B. Roux, T. Simonson, Implicit solvent models, Biophys. Chem. 78 (1–2) (1999) 1–20.

[68] K.A. Sharp, B. Honig, Calculating total electrostatic energies with the nonlinear Poisson–Boltzmann equation, J. Phys. Chem. 94 (19) (1990) 7684–7692.

[69] K.A. Sharp, B. Honig, Electrostatic interactions in macromolecules: theory and applications, Annu. Rev. Biophys. Biophys. Chem. 19 (1) (1990) 301–332.

[70] G.H. Shortley, R. Weller, Numerical solution of Laplace's equation, J. Appl. Phys. 9 (1938) 334–348.

[71] J. Strain, Tree methods for moving interfaces, J. Comput. Phys. 151 (1999) 616–648.

[72] M. Theillard, F. Gibou, T. Pollock, A sharp computational method for the simulation of the solidification of binary alloys, J. Sci. Comput. (2014).

[73] E.J.W. Verwey, J.T.G. Overbeek, K. Van Nes, Theory of the Stability of Lyophobic Colloids: The Interaction of Sol Particles Having an Electric Double Layer, Elsevier Publishing Company, 1948.

[74] Y.N. Vorobjev, H.A. Scheraga, A fast adaptive multigrid boundary element method for macromolecular electrostatic computations in a solvent, J. Comput. Chem. 18 (4) (1997) 569–583.

[75] J. Wang, C. Tan, Y.-H. Tan, Q. Lu, R. Luo, Poisson-Boltzmann solvents in molecular dynamics simulations, Commun. Comput. Phys. 3 (5) (2008) 1010–1031.

[76] A. Wiegmann, K.P. Bube, The explicit-jump immersed interface method: finite difference methods for pdes with piecewise smooth solutions, SIAM J. Numer. Anal. 37 (3) (2000) 827–862.

[77] S. Yu, W. Geng, G. Wei, Treatment of geometric singularities in implicit solvent models, J. Chem. Phys. 126 (24) (2007) 244108.

[78] H.-X. Zhou, Boundary element solution of macromolecular electrostatics: interaction energy between two proteins, Biophys. J. 65 (2) (1993) 955–963.

[79] Y. Zhou, S. Zhao, M. Feig, G. Wei, High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources, J. Comput. Phys. (ISSN 0021-9991) 213 (1) (2006) 1–30, https://doi.org/10.1016/j.jcp.2005.07.022, http://www.sciencedirect.com/science/article/pii/S0021999105003578.

[80] Z. Zhou, P. Payne, M. Vasquez, N. Kuhn, M. Levitt, Finite-difference solution of the Poisson–Boltzmann equation: complete elimination of self-energy, J. Comput. Chem. 17 (11) (1996) 1344–1351.