Journal of Computational Physics 230 (2011) 2125–2140

Contents lists available at ScienceDirect

Journal of Computational Physics

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

A second-order discretization of the nonlinear Poisson–Boltzmann equation over irregular geometries using non-graded adaptive Cartesian grids

Mohammad Mirzadeh [a,*], Maxime Theillard [a,b], Frédéric Gibou [a,c]

[a] Department of Mechanical Engineering, University of California, Santa Barbara, CA 93106-5070, USA
[b] Ecole Polytechnique, 91120 Palaiseau, France
[c] Department of Computer Science, University of California, Santa Barbara, CA 93106, USA

ARTICLE INFO

Article history:
Received 7 May 2010
Received in revised form 18 November 2010
Accepted 3 December 2010
Available online 10 December 2010

Keywords:
Nonlinear Poisson–Boltzmann equation
Non-graded adaptive grid
Octree data structure
Second-order discretization
Arbitrary geometries
Supercapacitors

ABSTRACT

In this paper we present a finite difference scheme for the discretization of the nonlinear Poisson–Boltzmann (PB) equation over irregular domains that is second-order accurate. The interface is represented by a zero level set of a signed distance function using Octree data structure, allowing a natural and systematic approach to generate non-graded adaptive grids. Such a method guaranties computational efficiency by ensuring that the finest level of grid is located near the interface. The nonlinear PB equation is discretized using finite difference method and several numerical experiments are carried which indicate the second-order accuracy of method. Finally the method is used to study the supercapacitor behaviour of porous electrodes.

1. Introduction

The Poisson–Boltzmann (PB) equation describes the electrostatic potential distribution around charged particles (colloids, macromolecules, membranes, etc.) in an electrolyte solution and thus it has a wide range of applications, from colloid science and micro-fluidics [25,18,29] to biochemistry and biophysics [19,26]. Due to its nonlinear nature, however, early studies of PB equations were mainly either concerned with the linearized limit, as in Debye–Hückel approximation, or limited to very simple geometries, as in Gouy–Chapman solution [25,18]. However, neither case really applies to biophysical systems, for example, where one should deal with highly charged and complicated macromolecules like DNA.

Various numerical techniques have been applied to both the linear and nonlinear PB equation during the past decades. Gilson et al. [9], Davis and McCammon [7], Nicholls and Honig [23] and Luo et al. [16] applied finite difference techniques to discretized the linearized PB equation in two and three spatial dimensions and discussed different choices of linear solver for their system. Finite difference method has also been successfully applied to the full nonlinear PB equation as seen in the works of Allison et al. [1], Jayaram et al. [13], Luty et al. [17] and Yang et al. [29]. Due to complicated geometries involved when applying the PB equation to biological systems, finite element methods have been developed to better capture the irregular interfaces [12,5,6,10,3]. Finite element methods have the nice property to create symmetric linear system which are cheaper to invert than non-symmetric ones. Furthermore, relying on *a posteriori* error estimates, it is possible to generate

* Corresponding author.
E-mail address: m.mirzadeh@engineering.ucsb.edu (M. Mirzadeh).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.12.008

adaptive grids which would even further increase the computational efficiency. One major drawback, however, is that elements need to conform to the boundaries and they must not be skewed in order to provide accurate and robust discretizations, grid generation could prohibitively become a bottleneck for very complicated geometries. This process could even become more complex and time consuming if one needs to explicitly track particles movement due to electrostatic forces.

Closely related to FEM, boundary element methods have also been used to solve the PB equation [30,14,24]. Compared to finite difference and finite element methods, the main advantage of boundary element methods lies in the fact that they essentially reduce the dimensionality of the problem by one. Although this is a desired property, as it would greatly reduce the size of linear system, one should note these type of methods typically tend to produce very dense linear systems which are expensive to invert. In addition, boundary integral methods are not straightforward to implement in three spatial dimensions. Finally we make a quick note regarding the application of the finite volume method as seen in the works of Holst and Saied [11].

In this paper we describe a finite difference approach to solve the full nonlinear PB equation over irregular geometries on Cartesian grids. To accomplish this goal, we represent the computational domain with an implicit function whose zero level set shall represent the irregular interface. The computational domain is thus implicitly captured without the need to explicitly fit the boundary. Since there are naturally two different length scales associated with the PB equation (see Section 2), it is desired to have an adaptive grid with the finest level of resolution close to the interface. We will be using Octree data structure as a natural way to generate adaptive Cartesian grid which will be used to discretize the PB equation.

2. Poisson–Boltzmann equation

Consider a particle or a surface that has a fixed surface electrostatic potential due to surface charges. When such a body is immersed inside an electrolyte, the electric field, due to the electrostatic potential, attracts the counter-ions and repels the co-ions in the solution to form a cloud of oppositely charged ions near the interface. The charged cloud, known as electric double layer (EDL), is known to consist of two layers, the *Stern layer* and the*diffuse layer*. While the Stern layer is believed to have the thickness of roughly one ionic diameter, the ion density in the diffuse layer, under equilibrium conditions, obeys the Boltzmann distribution [25,18], i.e.

$$n_i = n_0 \exp\left(\frac{-z_i e}{k_B T}\psi\right),$$

where ψ is the potential field in the charged cloud. Considering a symmetric $z{:}z$ electrolyte, the charge density inside the EDL may be found as:

$$\rho_e = ze(n_+ - n_-) = -2zen_0 \sinh\left(\frac{ze}{k_B T}\psi\right).$$

Finally the Poisson–Boltzmann equation may be obtained using this expression for the charge density in the Poisson equation for electrostatic potential field,

$$\nabla^2\psi = \frac{-\rho_e}{\epsilon} = \frac{2zen_0}{\epsilon}\sinh\left(\frac{ze}{k_B T}\psi\right).$$

The EDL thickness is typically measured in terms of the *Debye length* λ_D defined as:

$$\lambda_D = \sqrt{\frac{\epsilon k_B T}{2n_0 z^2 e^2}},$$

which is typically of the order of nanometers. Based on this, the PB equation may be non-dimensionalized by introducing the following variables:

$$x_i = L\tilde{x}_i \quad \nabla = L^{-1}\tilde{\nabla} \quad \psi = \frac{k_B T}{ze}\tilde{\psi} \quad \kappa = \frac{L}{\lambda_D},$$

where L is some characteristic length scale. The non-dimensional PB equation, ignoring the tildes, then becomes:

$$\nabla^2\psi = \kappa^2 \sinh(\psi). \tag{2.1}$$

It is easy to see that for high surface potentials, Eq. (2.1) becomes highly nonlinear and thus poses numerical difficulties near the interface, whereas far from interface the potential dies off exponentially. This, in part, demonstrates the need for an adaptive grid with a fine resolution near the interface. Holst and Saied [12], did a comprehensive study of treating the nonlinear term and concluded that damped inexact Newton's method is an efficient way to linearize the PB equation. In this paper, however, we will be using a slightly modified strategy where the solution at the previous step is used to expand the nonlinear term about, i.e. we write:

$$\nabla^2\psi^{n+1} = \kappa^2(\sinh(\psi^n) + (\psi^{n+1} - \psi^n)\cosh(\psi^n)) + \mathcal{O}((\psi^{n+1} - \psi^n)^2). \tag{2.2}$$

3. Numerical method

3.1. Grid generation

Consider the computational domain Ω along with its exterior boundary $\partial\Omega$ that is divided into two disjoint subdomains Ω^+ and Ω^- by a two-dimensional interface Γ. A level set function ϕ is used to represent different regions such that it is a signed distance function to the interface, i.e.,

$$\begin{cases} \phi > 0 & \text{in } \Omega^+, \\ \phi = 0 & \text{on } \Gamma, \\ \phi < 0 & \text{in } \Omega^- \end{cases}$$

with $|\nabla\phi| = 1$. We note, however, 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 cubes that are represented on an Octree data structure. Fig. 3.1 (left) illustrates a two-dimensional version of a computational domain along with its corresponding Quadtree (right). The grid generation starts by appointing the tree root, i.e. level zero, to the whole domain and recursively splitting every cell (level j) into four smaller cells (level $j+1$). This process is continued until either a certain resolution criterion is met or the tree has reached its maximum level. Following Min et al. [22] and Strain [28], one such criterion may be chosen as to divide a cell, with vertices in the set V, if the following inequality holds true:

$$\min_{v\in V} |\phi(v)| < \frac{LD}{2}. \tag{3.1}$$

Note that L is the Lipschitz constant of level set function ϕ and D is the diagonal size of the current cell.

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 directions. Alternatively a node is said to have a *T-junction* if it does not have a direct neighbor in at least one of the six possible directions. Note that a node can have at most one three-dimensional and one two-dimensional T-junctions (see Fig. 3.2).

3.2. Finite difference scheme

Let us rewrite Eq. (2.2) in the following form and assume a Dirichlet boundary condition at the interface Γ,

$$\begin{cases} (\nabla^2 - \kappa^2\cosh(\psi^n))\psi^{n+1} = \kappa^2(\sinh(\psi^n) - \psi^n\cosh(\psi^n)) & \text{in} \quad \Omega^-, \\ \psi = g(\vec{x}) & \text{on} \quad \Gamma. \end{cases} \tag{3.2}$$

To obtain a finite difference approximation to the Laplacian operator on the Octree mesh, we follow the works of Min et al. [22] and Chen et al. [4]. Fig. 3.2 depicts the most general configuration that may occur in the grid with at most one three-dimensional and one two-dimensional T-junction in negative y and negative x directions, respectively. To treat the T-junctions a linear interpolation is used to obtain the ghost values at the v_4 and v_5 nodes:

$$\psi_4^g = \frac{s_7\psi_8 + s_8\psi_7}{s_7 + s_8} \tag{3.3}$$

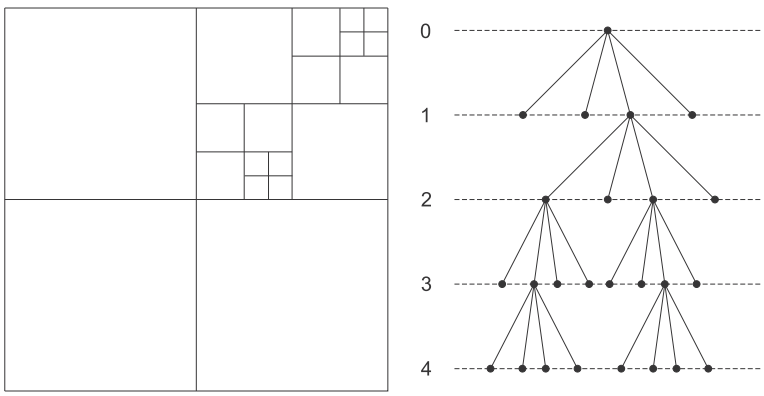


Fig. 3.1. Non-graded adaptive discretization of a two-dimensional domain (left) and its corresponding Quadtree (right).

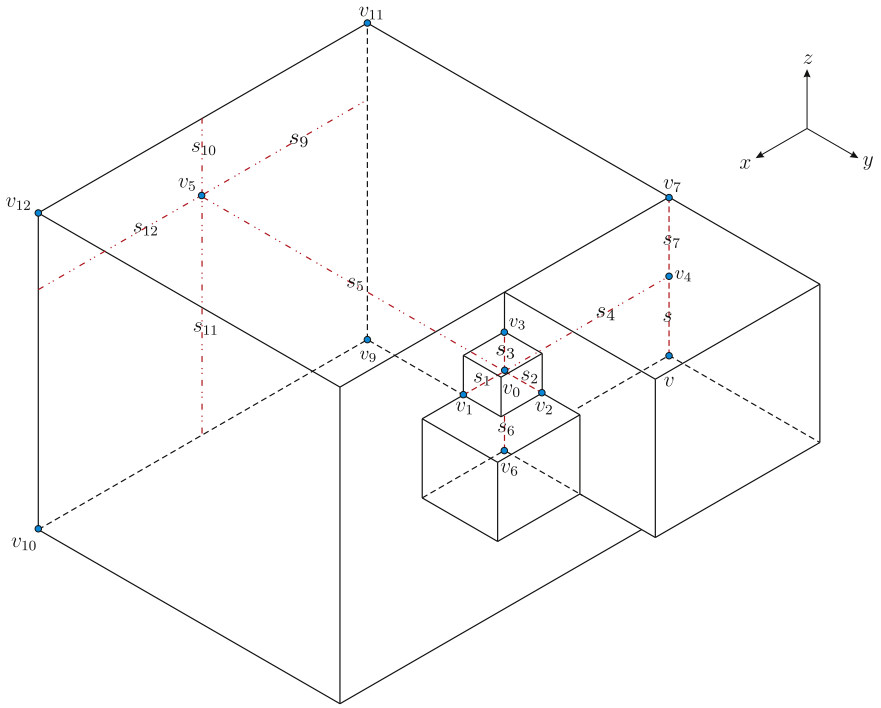


Fig. 3.2. A general configuration for a three-dimensional mesh. Node v_0 has a three-dimensional T-junction in negative y direction (called v_5) and a two-dimensional T-junction in negative x direction (called v_4).

and

$$\psi_5^g = \frac{s_{11}s_{12}\psi_{11} + s_{11}s_9\psi_{12} + s_{10}s_{12}\psi_9 + s_{10}s_9\psi_{10}}{(s_9 + s_{12})(s_{10} + s_{11})}. \tag{3.4}$$

A simple Taylor analysis shows that

$$\psi_4 = \psi_4^g - s_7 s_8 \psi_{zz}|_{v_0} + \mathcal{O}\left(h_1^3\right)$$

and

$$\psi_5 = \psi_5^g - s_9 s_{12}\psi_{xx}|_{v_0} - s_{10}s_{11}\psi_{zz}|_{v_0} + \mathcal{O}\left(h_2^3\right).$$

where $h_1 = \max\{s_4, s_7, s_8\}$ and $h_2 = \max\{s_5, s_9, s_{10}, s_{11}, s_{12}\}$. Note that to get second-order accuracy, it is first needed to eliminate the spurious $\psi_{zz}|_{v_0}$ and $\psi_{xx}|_{v_0}$ errors that are due to interpolation. Min et al. [22] showed that it is always possible to eliminate the spurious error terms by appropriate weighting of Laplacian operator as:

$$\nabla^2\psi_0 = \left(\frac{\psi_1 - \psi_0}{s_1} - \frac{\psi_0 - \psi_4^g}{s_4}\right)\frac{2\alpha}{s_4 + s_1} + \left(\frac{\psi_2 - \psi_0}{s_2} - \frac{\psi_0 - \psi_5^g}{s_5}\right)\frac{2}{s_5 + s_2} + \left(\frac{\psi_3 - \psi_0}{s_3} - \frac{\psi_0 - \psi_6}{s_6}\right)\frac{2\beta}{s_6 + s_3} + \mathcal{O}(h) \tag{3.5}$$

where $h = \max\{s_i\}$, ψ_4^g and ψ_5^g are obtained through Eqs. (3.3) and (3.4), respectively, and α and β are given by:

$$\alpha = 1 - \frac{s_{10}s_{11}}{s_2(s_2 + s_5)},$$
$$\beta = 1 - \frac{s_9 s_{12}}{s_5(s_2 + s_5)} - \alpha\frac{s_7 s_8}{s_4(s_1 + s_4)}.$$

Furthermore, they show that the finite difference matrix produced by Eq. (3.5) is an M-matrix provided that the anisotropic ratio of Octree is smaller than or equal to $\sqrt{2}$. The matrix is then non-singular if a Dirichlet boundary condition is imposed on at least one node.

3.3. Treatment of the boundary condition

Eq. (3.5) is used to discretize the PB equation in the Ω^- domain if all the corresponding neighboring and ghost nodes also reside in the same domain. If one of the nodes is located in the Ω^+ domain, however, this equation should be modified to take

into account the effect of the Dirichlet boundary conditions. In this paper we will only address the Dirichlet boundary condition on the interface Γ. The process is more easily understood in one spatial dimension and may easily be extended to three-dimensional domains using a dimension by dimension approach.

Consider a one-dimensional domain as in Fig. 3.3 where the computational domain is to the left of the interface Γ. Note that condition (3.1) always ensures a uniform grid across the interface and thus v_0 will always remain a regular node in two-dimensional and three-dimensional domains. ψ_{xx} in this case, may simply be approximated as

$$\psi_{xx} = \left(\frac{\psi_I - \psi_0}{s_I} - \frac{\psi_0 - \psi_1}{s_1}\right)\frac{2}{s_I + s_1} + \mathcal{O}(h), \tag{3.6}$$

where s_I is the distance to the interface and may be found via Taylor expansion of the level set function around v_0:

$$\phi(v_I) = \phi(v_0) + s_I\phi_x(v_0) + \frac{s_I^2}{2}\phi_{xx}(v_0), \tag{3.7}$$

where ϕ_x and ϕ_{xx} are simply given by:

$$\phi_x(v_0) = \frac{s_1\frac{\phi_4-\phi_0}{s_4} + s_4\frac{\phi_0-\phi_1}{s_1}}{s_1 + s_4},$$
$$\phi_{xx}(v_0) = \left(\frac{\phi_4 - \phi_0}{s_4} - \frac{\phi_0 - \phi_1}{s_1}\right)\frac{2}{s_1 + s_4}.$$

Finally, by definition we know that $\phi(v_I) = 0$ so that Eq. (3.7) may be solved for s_I as:

$$s_I = \begin{cases} \frac{-\phi_x(v_0) + \sqrt{\phi_x^2(v_0) - 2\phi_{xx}(v_0)\phi(v_0)}}{\phi_{xx}(v_0)} & \phi_{xx}(v_0) > \epsilon, \\ -\frac{\phi(v_0)}{\phi_x(v_0)} & |\phi_{xx}(v_0)| \leqslant \epsilon, \end{cases} \tag{3.8}$$

where ϵ is a small parameter to prevent division by zero.

3.4. Solution gradient

Here we make a quick note on the solution gradient in the field. Since the PB equation is only solved in the computational domain, i.e. in Ω^-, there is an ambiguity on the best way to compute the solution gradient, which here corresponds to the electric field $\vec{E} = -\nabla\phi$, near the interface. Aslam [2] first proposed a PDE approach to extrapolate the solution outside the computational domain and Min and Gibou [21] extended his work on quadtree/octree grids, which we will mention briefly here.

Suppose we are interested to extrapolate ψ from the computational domain Ω^- to Ω^+. One first needs to compute the quantity $\psi_{nn} = \vec{n}\cdot\nabla(\vec{n}\cdot\nabla\psi)$ in Ω^- and extrapolate it across the interface by solving:

$$\frac{\partial\psi_{nn}}{\partial\tau} + H(\phi,\psi_{nn})(\vec{n}\cdot\nabla\psi_{nn}) = 0,$$

where $H(\phi,\psi_{nn})$ is the Heaviside function defined below and τ is a fictitious time. Note that this is essentially equivalent to constant extrapolation of function ψ_{nn} across the interface. Next we define the quantity ψ_n in Ω^+ region such that its normal derivative is given by ψ_{nn}. This is accomplished by solving the PDE:

$$\frac{\partial\psi_n}{\partial\tau} + H(\phi,\psi_n)(\vec{n}\cdot\nabla\psi_n - \psi_{nn}) = 0.$$

Finally the solution, ψ, in the Ω^+ is found by enforcing its normal derivative to be equal to ψ_n through solving:

$$\frac{\partial\psi}{\partial\tau} + H(\phi,\psi)(\vec{n}\cdot\nabla\psi - \psi_n) = 0.$$

The Heaviside function $H(\phi,V)|_{v_i}$ is numerically set to zero if all the nodes involved in the computation of quantity V are in the computational domain, i.e. Ω^-. Therefore the Heaviside functions are defined as:

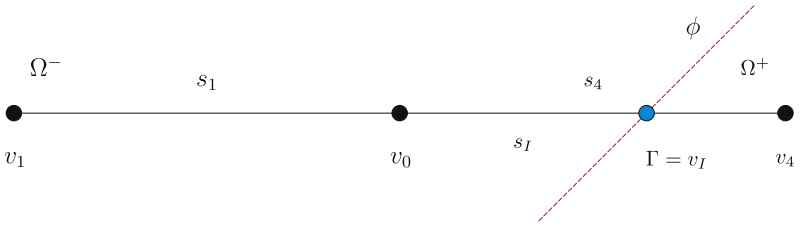


Fig. 3.3. The interface intersects the domain between nodes v_0 and v_4 at $\Gamma = v_I$.

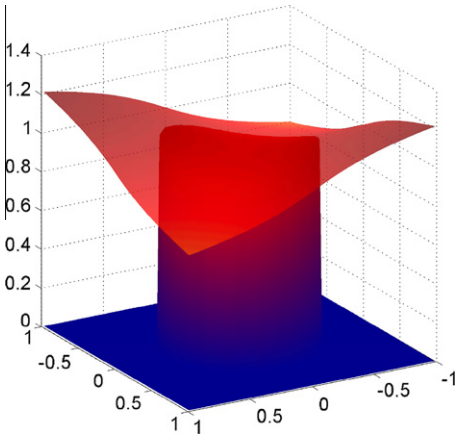


Fig. 3.4. Extrapolation procedure: The blue and red surfaces depict the solution before and after the extrapolation procedure, respectively. Note the smooth extension of the solution in Ω^+ on the red surface. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$H(\phi,\psi)|_{v_i} = \begin{cases} 0, & \text{if } \phi(v_i) < 0, \\ 1, & \text{otherwise}, \end{cases}$$

$$H(\phi,\psi_n)|_{v_i} = \begin{cases} 0, & \text{if } H(\phi,\psi)|_{v_i} = 0 \text{ for all } v_i \in ngbd(v_i), \\ 1, & \text{otherwise}, \end{cases}$$

$$H(\phi,\psi_{nn})|_{v_i} = \begin{cases} 0, & \text{if } H(\phi,\psi_n)|_{v_i} = 0 \text{ for all } v_i \in ngbd(v_i), \\ 1, & \text{otherwise}, \end{cases}$$

where $ngbd(v_i)$ denotes the set of direct neighboring nodes of the node v_i. One should refer to Min and Gibou [21] for more information on the discretization schemes used to solve the three PDEs. We conclude this section by providing an example of such extrapolation procedure as seen in Fig. 3.4. Note the smooth extension of the original solution (blue) in the Ω^+ region as shown by the red surface.

4. Examples in two spatial dimensions

In this section we will be considering examples in two spatial dimensions and show that our discretization scheme produces second-order accurate results in the L^1 and the L^∞ norms.

4.1. Example 1: circle

As for the first example, consider the computational domain $\Omega = [-1,1]^2$ that embeds a lower dimensional interface Γ represented by the zero level set of $\phi(x,y) = \sqrt{x^2+y^2} - 0.5$. Also assume that the exact solution for this example is given by $\psi_0(x,y) = e^{-xy}$. Thus we are interested in solving:

$$\begin{cases} \nabla^2\psi = \sinh(\psi) + f_0, \\ f_0 = \nabla^2\psi_0 - \sinh(\psi_0), \end{cases} \tag{4.1}$$

subject to a Dirichlet boundary condition given by the exact solution. Fig. 4.1 illustrates the interface along with the adaptive grid.

Table 4.1 reports the results obtained for this example and clearly shows the second-order accuracy of the method. Note that $res_{\min}$ and $res_{\max}$ represent the minimum and maximum grid resolution in each direction and "Grid points" is the total number of grid points in the domain.

4.2. Example 2: a two-dimensional spiky interface

As the second example, let us consider a two-dimensional spiky interface given by:

$$\phi = \min\{\phi_1, \phi_2\},$$
$$\phi_1 = \sqrt{|x| + 2|y|} - 0.9,$$
$$\phi_2 = \sqrt{2|x| + |y|} - 0.9,$$

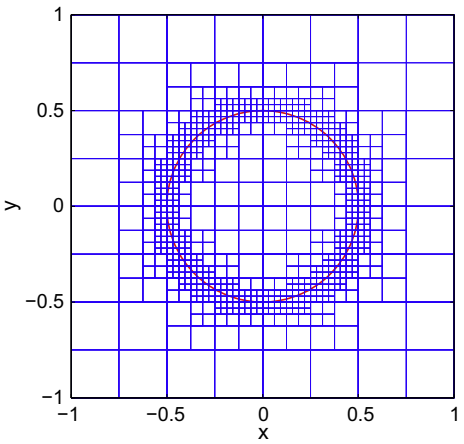


Fig. 4.1. A circular interface along with the adaptive grid.

Table 4.1
Accuracy analysis for example 1.

(res_{min}, res_{max})	Grid points	L^1 Error	Order	L^∞ Error	Order
(8,64)	861	1.054×10^{-5}	–	7.257×10^{-5}	–
(16,128)	1881	7.836×10^{-6}	0.428	7.055×10^{-5}	0.040
(32,256)	4345	1.881×10^{-6}	2.058	1.698×10^{-5}	2.095
(64,512)	10737	3.239×10^{-7}	2.537	2.844×10^{-6}	2.577
(128,1024)	29,689	4.924×10^{-8}	2.718	4.588×10^{-7}	2.632

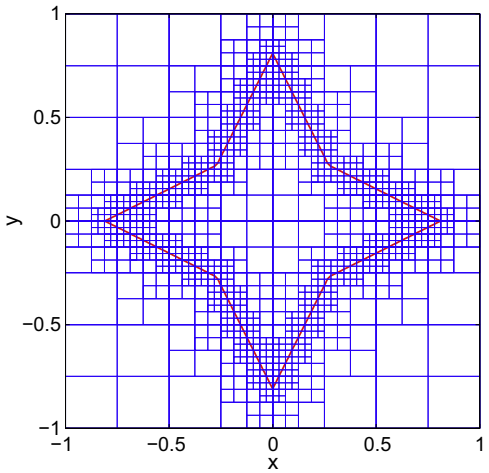


Fig. 4.2. A two-dimensional spiky interface along with the adaptive grid.

Table 4.2
Accuracy analysis for example 2.

(res_{min}, res_{max})	Grid points	L^1 Error	Order	L^∞ Error	Order
(8,64)	1753	5.521×10^{-6}	–	5.565×10^{-5}	–
(16,128)	3865	4.013×10^{-6}	0.460	5.414×10^{-5}	0.031
(32,256)	8509	1.251×10^{-6}	1.682	1.429×10^{-5}	1.516
(64,512)	18,897	2.190×10^{-7}	2.514	2.585×10^{-6}	2.352
(128,1024)	46,581	3.511×10^{-8}	2.641	3.985×10^{-7}	2.064

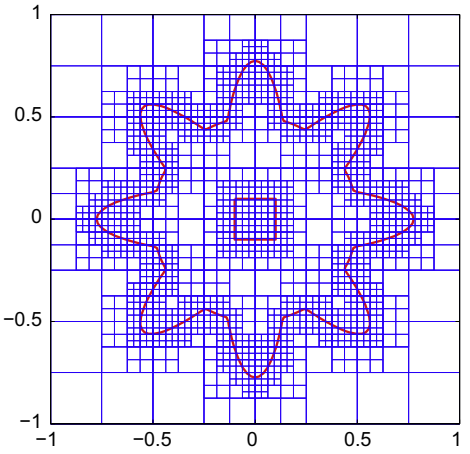


Fig. 4.3. An irregular interface along with the adaptive grid.

Table 4.3
Accuracy analysis for example 3.

(res_{min}, res_{max})	Grid points	L^1 Error	Order	L^∞ Error	Order
(8,64)	1753	3.499×10^{-6}	–	4.078×10^{-5}	–
(16,128)	3865	2.688×10^{-6}	0.380	3.988×10^{-5}	0.031
(32,256)	8509	1.132×10^{-6}	1.247	1.426×10^{-5}	1.516
(64,512)	18,897	2.241×10^{-7}	2.336	2.792×10^{-6}	2.352
(128,1024)	46,581	3.488×10^{-8}	2.684	6.675×10^{-7}	2.064

with the same exact solution as in example 4.1. Fig. 4.2 depicts the interface along with the computational grid. As seen in Table 4.2, even when the interface is not smooth, our finite difference scheme produces results that are second-order accurate.

4.3. Example 3: an irregular interface

Finally consider an irregular interface as demonstrated in Fig. 4.3 with same exact solution as given in example 4.1. Table 4.3 illustrates the accuracy analysis of this example.

5. Examples in three spatial dimensions

5.1. Example 1: single particles

As for the first example let us consider the case where $\Omega = [-1,1]^3$ where the exact solution is given by $\psi_0(x,y,z) = \sin(2\pi x)\sin(2\pi y)\sin(2\pi z)$. We are thus seeking a numerical approximation to the solution of Eq. (4.1) in three spatial dimensions subject to the Dirichlet boundary condition given by the exact solution. Two different types of interfaces are considered. Fig. 5.1 illustrates a sphere and a spiky three-dimensional surface where the level-set functions are given by:

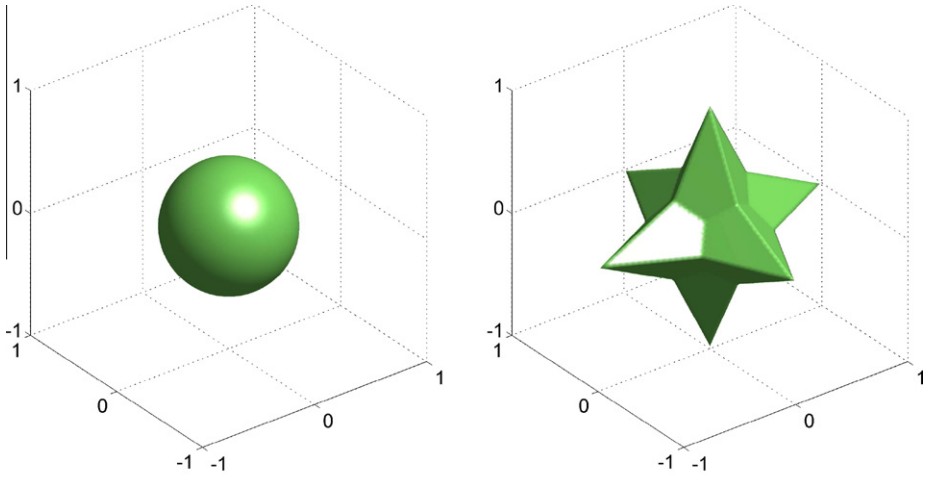
$$\phi_a = \sqrt{x^2 + y^2 + z^2} - 0.5$$

and

$$\begin{aligned}
\phi_b &= \min\{\phi_1, \phi_2, \phi_3\},\\
\phi_1 &= \sqrt{|x| + 2|y| + 2|z|} - 0.9,\\
\phi_2 &= \sqrt{2|x| + 2|y| + |z|} - 0.9,\\
\phi_3 &= \sqrt{2|x| + |y| + 2|z|} - 0.9,
\end{aligned}$$

respectively.

Tables 5.1 and 5.2 report the simulation results and provide an overall estimate for the order of accuracy of the method. It is easy to see that the numerical discretization is second-order accurate both in the L^1 and the L^∞ norms. Note that res_{min} and res_{max} are the minimum and the maximum resolution of the grid in one direction.



(a) A Sphere with a radius of 0.5 (b) A three dimensional spiky interface

Fig. 5.1. Interfaces used in the first example.

Table 5.1
Accuracy analysis for the sphere.

(res_{min}, res_{max})	Grid points	L^1 Error	Order	L^∞ Error	Order
(2,16)	2585	9.866×10^{-2}	–	9.612×10^{-1}	–
(4,32)	17,889	1.500×10^{-3}	6.039	6.502×10^{-3}	7.207
(8,64)	132,545	4.314×10^{-4}	1.798	1.751×10^{-3}	1.893
(16,128)	1,019,265	1.161×10^{-4}	1.894	4.605×10^{-4}	1.927

Table 5.2
Accuracy analysis for the rough three-dimensional surface.

(res_{min}, res_{max})	Grid points	L^1 Error	Order	L^∞ Error	Order
(2,16)	3793	1.012×10^{-2}	–	2.911×10^{-1}	–
(4, 32)	26,257	2.408×10^{-4}	5.393	1.339×10^{-3}	7.764
(8, 64)	194,593	6.285×10^{-5}	1.938	3.272×10^{-4}	2.033
(16, 128)	1,496,641	1.608×10^{-5}	1.966	8.071×10^{-5}	2.019

5.2. Example 2: electrostatic interactions between particles

As a second example, a domain consisting of multiple irregular particles is considered. Following the previous example, we consider the domain to be $\Omega = [-1,1]^3$ and the exact solution to be given by $\psi_0(x,y,z) = \sin(2\pi x)\sin(2\pi y)\sin(2\pi z)$. Fig. 5.2 illustrates the interface cross-section and its corresponding adaptive grid. In accordance with our previous results, Table 5.3 shows the convergence analysis that confirms second-order accuracy both in the L^1 and the L^∞ norms.

Since the level set function is used for grid generation and interface representation, it is an easy task to consider a domain consisting of several particles. This is very interesting and useful from a practical point of view as it allows for direct computation of electrostatic interactions between charged particles. As such, consider a domain $\Omega = [-1,1]^3$ where four of these particles are held fixed at specified locations. For such a system, we wish to solve the PB equation along with the following set of boundary conditions:

$$\begin{cases} \psi(\vec{x}) = 1 & \text{on } \Gamma, \\ \psi(\vec{x}) = 0 & \text{on } \partial\Omega. \end{cases} \tag{5.1}$$

Once the solution to the electric potential is known one may simply compute the electric field as $\vec{E} = -\nabla\psi$. By knowing the electric, it is possible to get the electrostatic forces acting on particles by integrating the electric stress tensor around the objects [25,18], i.e.

$$F_i = \int\int_S \sigma_{ij} n_j dA, \tag{5.2}$$

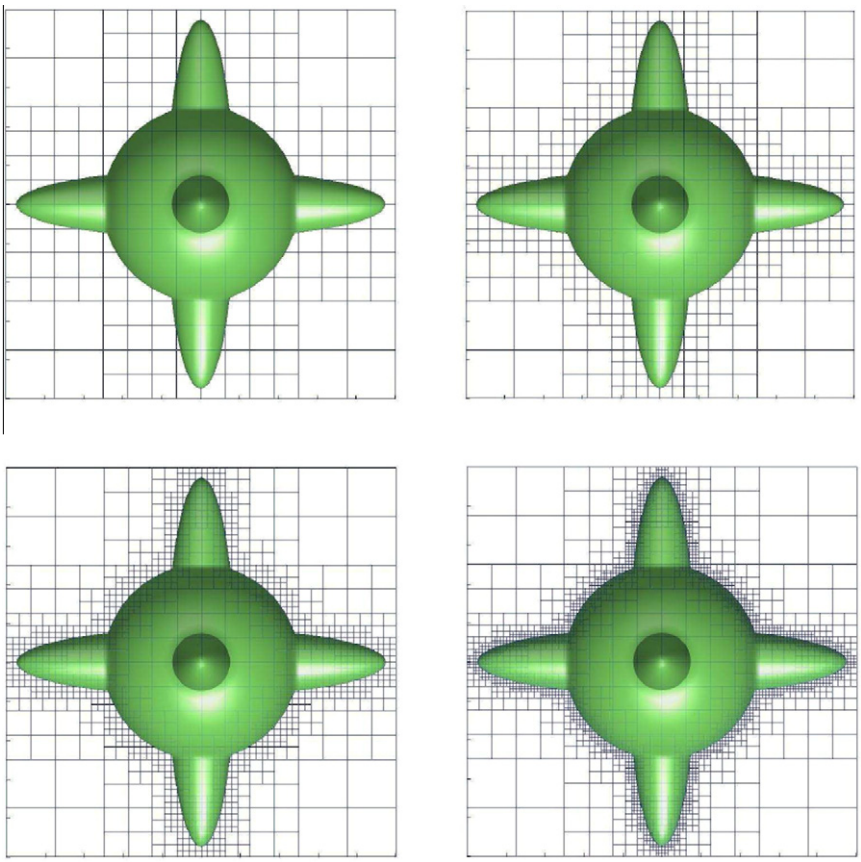


Fig. 5.2. Adaptive grid generation as $res_{\max}$ is increased.

Table 5.3
Accuracy analysis for example 2.

$(res_{\min}, res_{\max})$	Grid points	L^1 Error	Order	L^∞ Error	Order
(2,16)	3013	1.02×10^{-1}	–	1.14×10^{-0}	–
(4,32)	21,169	1.25×10^{-3}	6.36	6.51×10^{-3}	7.45
(8,64)	158,209	3.60×10^{-4}	1.79	1.75×10^{-3}	1.86
(16,128)	1,222,273	9.71×10^{-5}	1.89	4.60×10^{-4}	1.92

where σ_{ij} is given by:

$$\sigma_{ij} = -\Pi\delta_{ij} + \epsilon\left(E_i E_j - \frac{1}{2}\,|\,E|^2\delta_{ij}\right), \tag{5.3}$$

where Π is the osmotic pressure and ϵ is the dielectric constant of the electrolyte. Solution results are shown in Figs. 5.3 and 5.4. As indicated in Section 3.4, the electric field has been computed after extending the solution Ω^+ region, which in this case corresponds to the inside of particles. One may refer to Min and Gibou [20] for a reference on performing the surface integral in Eq. (5.2).

5.3. Example 3: surface roughness

As the last problem, let us consider the interface to be a rough boundary, for which $\Omega = [0,1]^3$ and the level set function is given by:

$$\phi = \delta(1 + \sin(2\pi x)\sin(2\pi y)) - z,$$

where we consider $\delta = 0.15$. This might be used to model the effects of surface roughness on the slip velocity produced in an electro-osmotic flow. Fig. 5.5 illustrates three different cross-sections of the interface along with the adaptive grid. Before

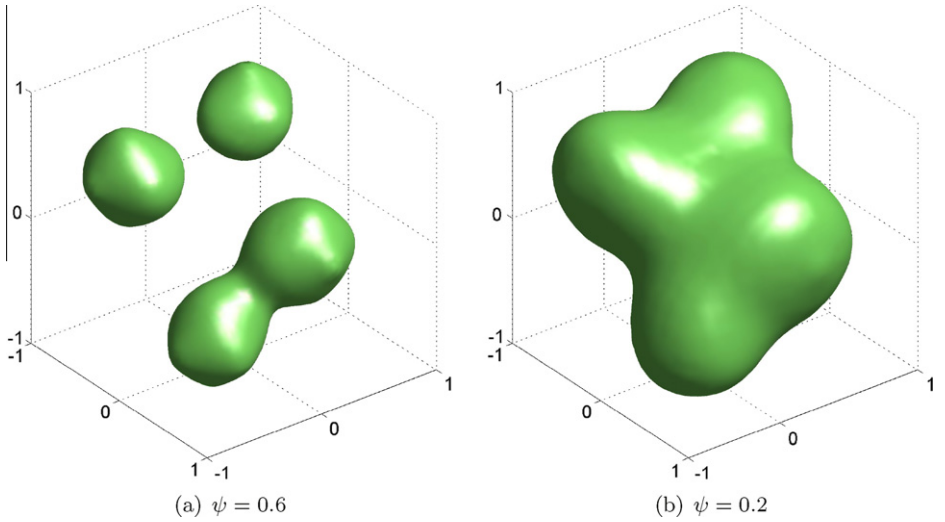


Fig. 5.3. Isosurfaces of electrostatic potential ψ.

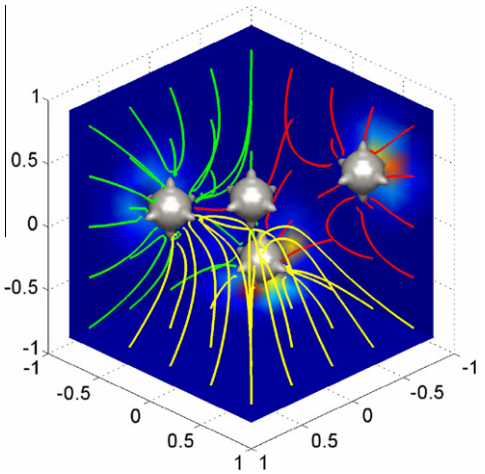


Fig. 5.4. Electric fields. Each set of colored field line originates from a different face. Background faces are colored in term of the electric field magnitude.

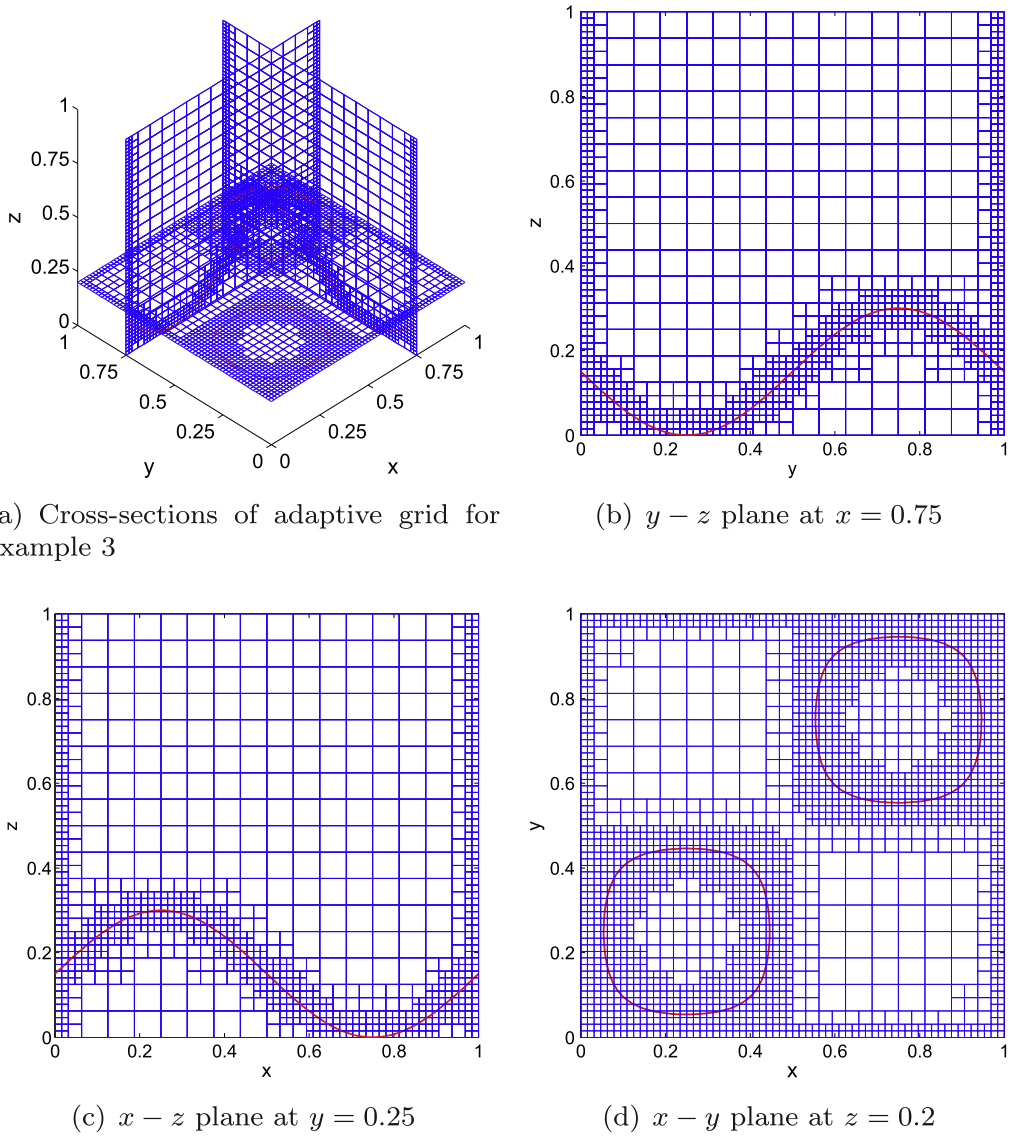
considering the real boundary condition, and to analyze the accuracy of the method, let us assume that the exact solution is given by $\psi_0 = \sin(2\pi x)\sin(2\pi y)e^{-z}$. Table 5.4 reports the results obtained in this case. It is easy to see that our method produces second-order accurate results. Next, let us consider the solution to the PB equation under the following set of boundary conditions,

$$\begin{cases} \psi(x,y,z) = \zeta_0 & \text{on the surface,} \\ \psi(x,0,z) = \psi(x,1,z) & \text{i.e periodic in } x \text{ direction,} \\ \psi(0,y,z) = \psi(1,y,z) & \text{i.e periodic in } y \text{ direction,} \\ \psi(x,y,z) \sim \alpha e^{-\kappa z} & z \to \infty, \end{cases} \tag{5.4}$$

where ζ_0 is the surface ζ-potential. Note that the "physical" boundary condition when $z \to \infty$ is simply $\psi = 0$. However, since $\psi(x,y,z \to \infty) \ll 1$, it is possible to get the asymptotic behavior of the potential for large values of z as given in Eq. (5.4). When applying this type of boundary condition numerically, the coefficient α is found by a simple iterative procedure.

In general there is no analytical solution for the PB equation for an arbitrary surface amplitude, unless $\delta = 0$, where the exact solution is given by Eq. (5.5) and may be used to check the accuracy of our method. Table 5.5 also confirms our previous results in that our discretization is second-order accurate.

$$\psi = 4\tanh^{-1}\left(\tanh\left(\frac{\zeta_0}{4}\right)e^{-z}\right). \tag{5.5}$$



(a) Cross-sections of adaptive grid for example 3

(b) $y - z$ plane at $x = 0.75$

(c) $x - z$ plane at $y = 0.25$

(d) $x - y$ plane at $z = 0.2$

Fig. 5.5. Three-dimensional adaptive grid and its cross-sections around the surface roughness.

Table 5.4
Accuracy analysis for the roughness problem.

(res_{min}, res_{max})	Grid points	L^1 Error	Order	L^∞ Error	Order
(8,16)	3906	3.36×10^{-3}	–	1.01×10^{-2}	–
(16, 32)	27,107	7.37×10^{-4}	2.19	2.66×10^{-3}	1.92
(32,64)	200,805	1.76×10^{-4}	2.07	7.06×10^{-4}	1.91

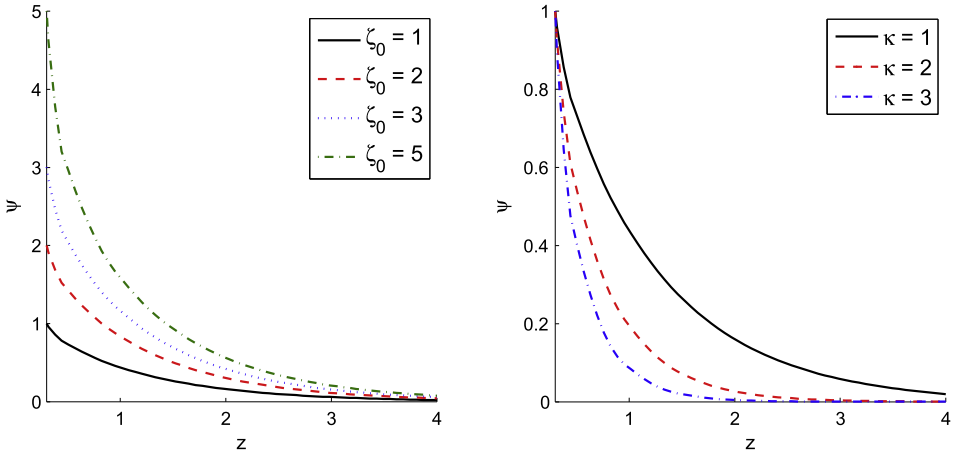
Finally it is interesting to investigate the effects of the surface potential, ζ_0, and the EDL thickness, κ^{-1}, for a non-flat surface i.e. $\delta \neq 0$. Figs. 5.6(a) and (b) show the effect of these parameters on the potential drop across the EDL. The simulation was done on an adaptive grid with $(res_{min}, res_{max}) = (16,64)$.

5.4. Remarks on the iteration scheme

The Poisson–Boltzmann equation is nonlinear and in this paper we have used the Newton's iteration method to handle the nonlinearity in an efficient fashion. It is thus interesting to study the dependence of various physical parameters, such as

Table 5.5
Accuracy analysis for the roughness problem.

(res_{min}, res_{max})	Grid points	L^1 Error	Order	L^∞ Error	Order
(8,16)	3394	1.57×10^{-4}	–	4.00×10^{-4}	–
(16,32)	22,819	3.91×10^{-5}	2.00	1.24×10^{-4}	1.69
(32,64)	165,733	9.82×10^{-6}	1.99	3.28×10^{-5}	1.92



(a) $\kappa = 1$ (b) $\zeta_0 = 1$

Fig. 5.6. Effects of surface potential and EDL thickness on the solution. Plots generated for $x = y = 0.25$ and $\delta = 0.15$.

the surface potential and the Debye layer thickness, as well as the grid resolution on the convergence of this iteration scheme. We note that physical parameters in general may have an effect on the 'strength' of the nonlinearity, whereas grid resolution is a typical numerical parameter that can influence the convergence of numerical methods. The stopping criterion we use for convergence is to consider that the difference between two consecutive iterate is less than a given tolerance, taken here to be 10^{-6}:

$$\max_{x \in \Omega^-} \|\psi^{v+1} - \psi^v\| < 10^{-6}.$$

Two examples are considered: the sphere described in Section 5.1 and the irregular surface described in Section 5.2. In both cases, a range of surface potential from $\zeta = 1$ to $\zeta = 5$ was considered for the Dirichlet boundary condition at the interface. For each of those cases, the problem was solved in the thick ($\kappa = 1$) and thin ($\kappa = 5$) double layer limit. In each of those cases, we considered a coarse, i.e. $(res_{min}, res_{max}) = (32, 128)$, and a fine, i.e. $(res_{min}, res_{max}) = (32, 512)$, computational grid. Tables 5.6 and 5.7 illustrate the variations in the number of iterations needed for convergence for the sphere and irregular surface, respectively.

Table 5.6
Number of iterations needed for convergence for the sphere example (cf. Section 5.1) as a function of different physical and numerical parameters.

Surface potential (ζ)	1.0	2.0	3.0	4.0	5.0
Coarse grid-Thick double layer	4	4	5	5	6
Fine grid-Thick double layer	4	4	5	5	6
Coarse grid-Thin double layer	3	4	4	4	5
Fine grid-Thin double layer	3	4	4	4	5

Table 5.7
Number of iterations needed for convergence for the irregular surface example (cf. Section 5.2) as a function of different physical and numerical parameters.

Surface potential (ζ)	1.0	2.0	3.0	4.0	5.0
Coarse grid-Thick double layer	4	4	5	5	6
Fine grid-Thick double layer	4	4	5	5	6
Coarse grid-Thin double layer	4	4	4	5	5
Fine grid-Thin double layer	4	4	4	5	5

Both results suggest that the scheme is efficient. The iteration scheme depends mainly on the surface potential used as the interface's boundary condition. This is expected since, as the surface potential increases in magnitude, the nonlinearity of the problem increases as well due to the sinh (ψ) term. The other physical parameters as well as the grid resolution, however, do not have a strong effect on the iteration number as they are not involved in the linearization approximation of the Poisson–Boltzmann equation. Needless to say, these parameters do play a role in the computational resources needed for each iteration.

6. Applications in modeling supercapacitors

In this section we will consider an important physical application of solving the nonlinear Poisson–Boltzmann equation in modeling the so-called supercapacitor behavior of porous materials: An electric double layer essentially behaves like a capacitor, where the charge is stored in the electric double layer. Unlike standard capacitors where the separation distance between plates is often of the order of micrometer or millimeter, the distance in the case of the electric double layers is of the order of nanometer. As a result, electric double layers tend to have very high capacitance per unit area ($\sim$10 $\mu F/cm^2$). The supercapacitor concept then refers to the use of highly porous materials as electrodes that can provide large relative surface areas. For example, electrodes made out of active carbon have been reported with surface areas up to 3000 m^2 per gram, resulting in total capacitance of $\sim$300 Farads per gram [8]. Due to their high capacitance for storing electrical charge and power, these types of capacitors hold a promise for many applications, such as in Hybrid Electric Vehicles (HEV). We do not elaborate this in more detail here and refer the interested reader to Kötz [15] and Simon and Gogotsi [27] for a review on supercapacitors and their applications.

To model such capacitors, we have considered a porous electrode made out of removing spherical sites from a periodic cubic cell. The spherical sites reside on a hexagonal lattice known as the Hexagonal Closed Packing (HCP) model, as illustrated schematically in Fig. 6.1. In this figure the black and red circles represent the location of spherical sites for two different lattice layers in the z direction (perpendicular to the page) and the green line represents the boundary of the computational domain. The lattice spacing, d_0, and site radius, r, are chosen in a way that the resulting object is porous and connected in all three spacial directions. It is easy to show that these criteria are met if the following condition is satisfied:

$$1 < \frac{2r}{d_0} < \sqrt{\frac{3}{2}}.$$

Using this condition, it is then possible to describe the entire geometry via a single parameter $0 < \alpha < 1$ defined such that:

$$\frac{2r}{d_0} = 1 + \alpha\left(\sqrt{\frac{3}{2}} - 1\right).$$

Figs. 6.2(a) and (b) illustrate the geometry obtained for $\alpha = 0.4$ and the iso-surfaces of the electric potential for $\zeta = 5$, respectively.

To compute the capacitance of the electrode, one first needs to compute the surface charge according to

$$q = \int_\Gamma \epsilon \nabla\psi \cdot \hat{n} d\mathcal{A}.$$

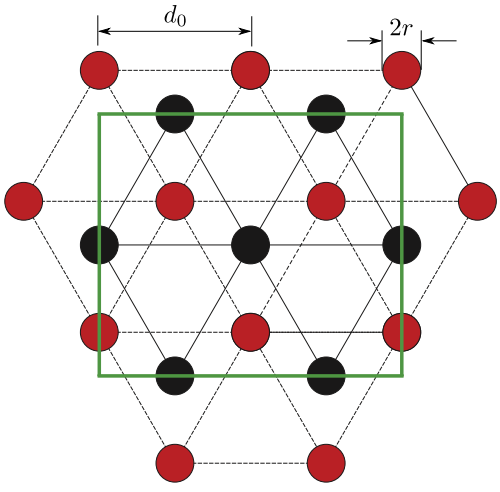
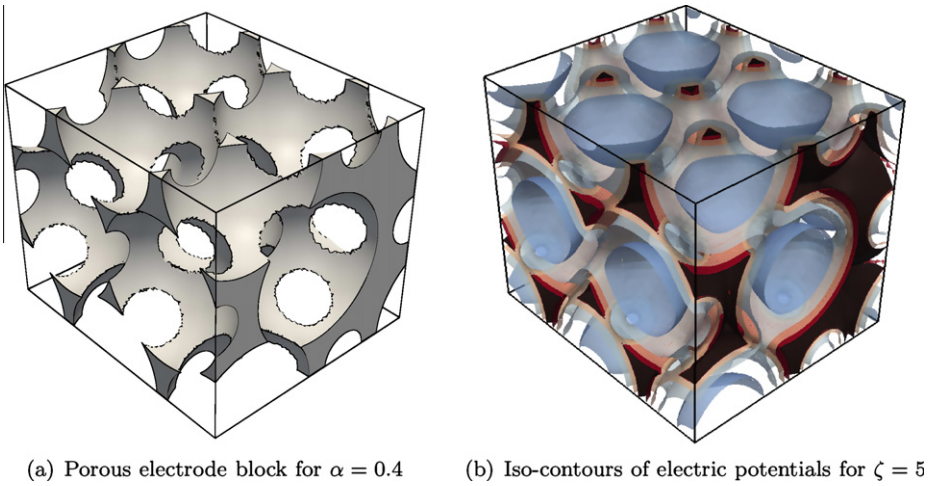


Fig. 6.1. Schematic of a Hexagonal Closed Packing (HCP) model of a porous electrode.



(a) Porous electrode block for $\alpha = 0.4$ (b) Iso-contours of electric potentials for $\zeta = 5$

Fig. 6.2. Generated porous electrode (left) and the potential iso-surfaces (right) of Section 6.

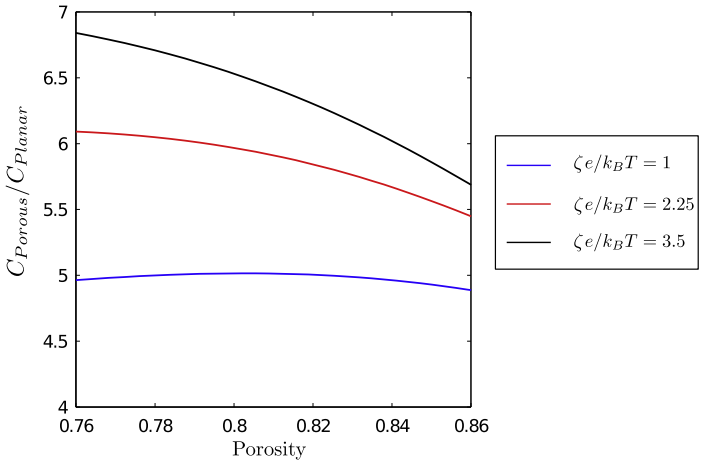


Fig. 6.3. Dependence of the electrode capacitance on the porosity.

Note that to accurately compute the surface charge, one first needs to compute the electric field as explained in Section 3.4. Next, the capacitance is obtained by solving the problem for a range of surface ζ potential and computing the following derivative:

$$C = \frac{\partial q}{\partial \zeta}.$$

Through this procedure, it is possible to study the effect of the electrode porosity on the capacitance. This is illustrated in Fig. 6.3 for range of surface potential from $\zeta = 1$ to $\zeta = 3$ and the double layer thickness $\kappa = 4.5$. In this figure the vertical axis represents the porous electrode capacitance normalized by the capacitance of a planar electrode having the same frontal area ($\mathcal{A}_{xy}$). The horizontal axis represents the porosity of the material, defined as the void volume fraction to the total cell volume. From this figure, it is easy to recognize the important role of the porosity in increasing the electrode capacitance. As expected, this effect is magnified as the surface potential is increased due to the nonlinear crowding of ions in the electric double layer. Finally we note that as the porosity of the material is increased, the capacitance is decreased. This phenomenon is only dependent on the behavior of the model used to describe the porous electrode and, in our case, may be attributed to the fast decrease of the surface area as the porosity is increased.

7. Concluding remarks

In this paper we have presented an efficient finite difference discretization for the Poisson–Boltzmann equation on irregular domains. The nonlinear equation is first approximated using the Newton's method and then discretized on a non-graded

adaptive Cartesian grids. Level set functions were used to represent the irregular domain, which notably facilitate the task of grid generation as well as handling multiple particles and objects. We have considered several different test cases that demonstrate that our method is second-order accurate in the L^∞ and the L^1 norms. Furthermore, the use of adaptive Cartesian grids with no constraint on the ratio between adjacent cells, produces a method that is efficient and versatile, allowing the robust discretization of virtually any irregular interfaces. Finally, we have considered an application of our method to the study of modeling porous electrode supercapacitors, considering a model based on the Hexagonal Closed Packing (HCP) crystal for the porous geometry. Our preliminary results clearly support the fact that the electrode capacitance is increased for porous materials.

Acknowledgment

The research of M. Mirzadeh, M. Theillard and F. Gibou was supported in part by the Office of Naval Research N00014-11-1-0027, the Department of Energy under Grant agreement DE-FG02-08ER15991, the National Science Foundation under Grant agreement DMS 0713858 and by the Institute for Collaborative Biotechnologies through contract No. W911NF-09-D-0001 from the US Army Research Office. The authors greatly acknowledge helpful discussions with Todd M. Squires on the physical aspect of the Poisson–Boltzmann equation and suggesting the modeling of supercapacitors.

References

[1] S.A. Allison, J.J. Sines, A. Wierzbicki, Solutions of the full Poisson–Boltzmann equation with application to diffusion-controlled reactions, J. Phys. Chem. 93 (1989) 5819.
[2] T.D. Aslam, A partial differential equation approach to multidimensional extrapolation, J. Comput. Phys. 193 (2004) 349.
[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 (2000) 1343.
[4] H. Chen, C. Min, F. Gibou, A supra-convergent finite difference scheme for the poisson and heat equations on irregular domains and non-graded adaptive cartesian grids, J. Sci. Comput. 31 (2007) 19.
[5] C.M. Cortis, R.A. Friesner, An automatic three-dimensional finite element mesh generation system for the Poisson–Boltzmann equation, J. Comput. Chem. 18 (1997) 1570.
[6] C.M. Cortis, R.A. Friesner, Numerical solution of the Poisson–Boltzmann equation using tetrahedral finite-element meshes, J. Comput. Chem. 18 (1997) 1591.
[7] M.E. Davis, J.A. McCammon, Solving the finite difference linearized Poisson–Boltzmann equation: a comparison of relaxation and conjugate gradient methods, J. Comput. Chem. 10 (1989) 386.
[8] E. Frackowiak, Carbon materials for the electrochemical storage of energy in capacitors, Carbon 39 (2001) 937.
[9] M.K. Gilson, K.A. Sharp, B.H. Honig, Calculating the electrostatic potential of molecules in solution: method and error assessment, J. Comput. Chem. 9 (1988) 327.
[10] M. Holst, N. Baker, F. Wang, Adaptive multilevel finite element solution of the Poisson–Boltzmann equation I. Algorithms and examples, J. Comput. Chem. 21 (2000) 1319.
[11] M. Holst, F. Saied, Multigrid solution of the Poisson–Boltzmann equation, J. Comput. Chem. 14 (1993) 105.
[12] M.J. Holst, F. Saied, Numerical solution of the nonlinear Poisson–Boltzmann equation: developing more robust and efficient methods, J. Comput. Chem. 16 (1994) 337.
[13] B. Jayaram, K.A. Sharp, B. Honig, The electrostatic potential of b-DNA, Biopolymers 28 (1989) 975.
[14] A.H. Juffer, E.F.F. Botta, B.A.M. van Keulen, A. van der Ploeg, H.J.C. Berendsen, The electric potential of a macromolecule in a solvent: a fundamental approach, J. Comput. Phys. 97 (1991) 144.
[15] R. Kötz, Principles and applications of electrochemical capacitors, Electrochim. Acta 45 (2000) 2483.
[16] R. Luo, L. David, M.K. Gilson, Accelerated Poisson–Boltzmann calculations for static and dynamic systems, J. Comput. Chem. 23 (2002) 1244.
[17] B.A. Luty, M.E. Davis, J.A. McCammon, Solving the finite-difference non-linear Poisson–Boltzmann equation, J. Comput. Chem. 13 (1992) 1114.
[18] J. Lyklema, Fundamentals of Interface and Colloid Science, vol. 2, Academic Press, 1995.
[19] S. McLaughlin, The electrostatic properties of membranes, Ann. Rev. Biophys. Biophys. Chem. 18 (1989) 113.
[20] C. Min, F. Gibou, Geometric integration over irregular domains with application to level-set methods, J. Comput. Phys. (2007) 1432.
[21] C. Min, F. Gibou, A second order accurate level set method on non-graded adaptive cartesian grids, J. Comput. Phys. 225 (2007) 300.
[22] C. Min, F. Gibou, H.D. Ceniceros, A supra-convergent finite difference scheme for the variable coefficient poisson equation on non-graded grids, J. Comput. Phys. 218 (2006) 123.
[23] A. Nicholls, B. Honig, A rapid finite difference algorithm, utilizing successive over-relaxation to solve the Poisson–Boltzmann equation, J. Comput. Chem. 12 (1991) 435.
[24] A.A. Rashin, J. Malinsky, New method for the computation of ionic distribution around rod-like polyelectrolytes with the helical distribution of charges. I. General approach and a nonlinearized Poisson–Boltzmann equation, J. Comput. Chem. 12 (1991) 981.
[25] W.B. Russel, D.A. Saville, W.R. Schowalter, Colloidal Dispersions, Cambridge University Press, 1989.
[26] K.A. Sharp, B. Honig, Electrostatic interactions in macromolecules: theory and applications, Ann. Rev. Biophys. Biophys. Chem. 19 (1990) 301.
[27] P. Simon, Y. Gogotsi, Materials for electrochemical capacitors, Nature Mater. 7 (2008) 845.
[28] J. Strain, Tree methods for moving interfaces, J. Comput. Phys. 151 (1999) 616.
[29] R.-J. Yang, L.-M. Fu, Y.-C. Lin, Electroosmotic flow in microchannels, J. Colloid Interface Sci. 239 (2001) 98.
[30] B.J. Yoon, A.M. Lenhoff, A boundary element method for molecular electrostatics with electrolyte effects, J. Comput. Chem. 11 (1990) 1080.