UNIVERSITY OF CALIFORNIA
Santa Barbara

Discretization of Poisson-Boltzmann and Poisson-Nernst-Planck Equations with Applications to Electrochemical Systems

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Mechanical Engineering

by

Seyed Mohammad Mirzadeh

Committee in Charge:

Professor Frederic Gibou, Chair

Professor Todd Squires

Professor Sumita Pennathur

Professor Linda Petzold

September 2014

The Dissertation of
Seyed Mohammad Mirzadeh is approved:

Professor Todd Squires

Professor Sumita Pennathur

Professor Linda Petzold

Professor Frederic Gibou, Committee Chairperson

June 2014

Discretization of Poisson-Boltzmann and Poisson-Nernst-Planck Equations with Applications to Electrochemical Systems

by

Seyed Mohammad Mirzadeh

To my mom, Ensi, whose support and love made this work possible.

Acknowledgements

Ever since I was a child I had the dream of becoming a scientist. In this never-ending journey, studying toward a doctorate degree has always been an important goal of mine and now that I have succeeded, it is only fair that I pay my gratitude to all those who helped me along the way and gave me hope and support.

First and foremost I thank my advisors, Frederic Gibou and Todd Squires. Frederic, your support for doing diverse research is phenomenal and absolutely essential to my success here in UCSB. Thank you for giving me freedom and encouraging me to always think of new approaches to problems. This has proved time and again helpful in my Ph.D. work; like the countless discussions we had on removing the fictitious fluxes when solving the PNP equations and eventually ended up devising something completely new and effective. I should also thank you for your support whenever I was going through difficulties in my research, always giving me hope that I will succeed. You are a wonderful person and anyone is fortunate to have you as their advisor.

Todd, this work would have not been possible without your help. Had I not taken your class on "Asymptotic Analysis" in my first year, I would have probably never knew about you. But I did! and soon realized I would love to do research with you. You are a great scientist and a phenomenal teacher. I believe that, without exaggeration, your classes are amongst the best learning experiences I have ever had during college and graduate school. Thank you so much for all those wonderful discussions we had in during your classes or in your office. I will never forget the impact you have had on me in my career.

I would also like to thank other members of my committee, Linda Petzold and Sumita Pennathur, with whom I have had several fruitful discussions at different stages of my Ph.D. work. Thank you so much for your support.

My success here in UCSB is rooted in my strong background that was formed when I was an undergraduate student. I am grateful to all those wonderful people back home at University of Tehran. Specifically I'd like to thank my undergraduate advisor, Keyvan Sadeghy, who first made me interested in Fluid Mechanics. I would also like to thank Vahid Esfahanian for his wonderful class on CFD that first made me interested in computational methods. Without them I would have probably chosen a different path in my research career. Finally, I would like to thank Abdolhamid Hannaneh for being such a great scientist, phenomenal teacher, and a wonderful human being. I will never forget your memory and I am forever thankful for all the things I have learned from you.

I would like to thank all my friends, here in UCSB and back home. Specifically I would like to thank Mohamad Mehdi Nasr-Azadani, Arthur Guittet, Carsten Burstedde, Brian Giera, Maxime Theillard, Miles Detrixhe, Ali Nabi, and Andy Pascal for many fruitful discussions that we have had in the past. I would also like to thank all the present and past members of CASL and Squires groups who have created such a diverse and wonderful place for research. Finally, I'd like to thank my friends Mohamad Mehdi Nasr-Azadani, Ida Karimfazli, Omid Hejazi, Mahyar Valizadeh, Babak Gholami, Atefeh Kazeroonian, Amir Farzad, Behzad Nateq, and Hasan Mostajab-o-Aldaveh for their individual role in shaping the person I am today at different stages of my life.

Last, but not least, I should thank my family. Thank you to my sisters Monir, Mahin, Mohtaram, and Mahboobeh and my brothers Mostafa and Mehdi for being supportive. Thank you to my father, Hashem, who always encouraged me in fulfilling my dreams of becoming a scientist. Your memories will always remain with me and your wisdom will guide me in hard times. Finally, thank you to my mother, Ensi, who has done nothing short of supporting me since I was a child. I will never forget your kindness in my life.

Seyed Mohammad Mirzadeh,

Santa Barbara, CA, June 2014.

Curriculum Vitæ

Seyed Mohammad Mirzadeh

Education

2014 Ph.D. in Mechanical Engineering, University of California, Santa Barbara, California, USA.

2008 B.S. in Mechanical Engineering, University of Tehran, Tehran, Iran.

Experience

2008 – 2014 Graduate Research Assistant, University of California, Santa Barbara.

2008 – 2014 Teaching Assitant, University of California, Santa Barbara. Duties at various times have included office hours and paper grading for several undergraduate courses related to *Fluid Mechanics*, *Heat Transfer*, *Thermodynamics*, and *Engineering Mathematics.*

2010 – 2014 Reviewer for *Journal of Computational Physics*, *SIAM Journal on Scientific Computing*, *Physical Chemistry Chemical Physics*, and *Physics of Fluids journals.*

Selected Publications

- **Mirzadeh, M.**, Theillard, M., and Gibou, F., *Journal of Computational Physics*, 230 (2011), 2125-2140.

- **Mirzadeh, M.**, Theillard, M., Helgadottir, A., Boy, D., and Gibou, F., *Communications in Computational Physics*, 13 (2013), 150-173

- Nabi, A., **Mirzadeh, M.**, Gibou, F., and Moehlis, J., *Journal of Computational Neuroscience*, 34, 259-271 (2013).

- Nabi, A., **Mirzadeh, M.**, Gibou, F., and Moehlis, J., *Proceedings of the 2012 American Control Conference*, 4751-4756.

- **Mirzadeh, M.** and Gibou, F., *Journal of Computational Physics*, 274 (2014), 633-653.

- **Mirzadeh, M.**, Gibou, F., and Squires T.M., *Physical Review Letters*, 113, 097701, (2014).

Conference Presentations

- **Mirzadeh, M.**, Squires, T.M. and Gibou, F. "An efficient discretization of the Poisson-Boltzmann equation with applications to electrostatic force calculation" 63^{rd} Annual Meeting of the APS Division of Fluid Dynamics (2010). Long Beach, California.

- **Mirzadeh, M.**, Squires, T.M., and Gibou, F. "Computational Modeling of Porous Supercapacitors Using Non-graded Adaptive Cartesian Grids" SIAM Conference on Computational Science and Engineering (2011). Reno, Nevada.

- **Mirzadeh, M.**, Gibou, F., and Squires, T.M. “Pore-scale Computational Modeling Of Porous Supercapacitors” 48th Annual Technical Conference of Society of Engineering Sciences (2011). Evanston, Illinois.

- Paustian, J.S., **Mirzadeh, M.**, Kim, D.H., Khair, A.S., Gibou, F., Chmelka, B.F. and Squires, T.M. “Self-assembly of nano-scale porous structures within microfabricated fluidic systems for sample preconcentration and energy storage” *poster session at* ICB Army-Industry Collaboration Conference (2012). Santa Barbara, California.

- **Mirzadeh, M.**, Squires, T.M., and Gibou, F. “Direct Numerical Simulation of Charging Dynamics of Porous Electrodes” SIAM Conference on Computational Science and Engineering (2013). Boston, Massachusetts.

- **Mirzadeh, M.**, Guittet, A., Burstedde, C., and Gibou, F. “Tree-based Adaptive Level-set Methods on Highly Parallel Machines” SIAM Conference on Parallel Processing for Scientific Computing (2014). Portland, Oregon.

Abstract

Discretization of Poisson-Boltzmann and Poisson-Nernst-Planck Equations with Applications to Electrochemical Systems

Seyed Mohammad Mirzadeh

In this work we present numerical methods that are suitable for studying a variety of electrochemical systems, such as charging kinetics of porous electrodes used in energy storage devices (e.g. supercapacitors, batteries, or fuel cells), calculation of binding energies in biomolecules, and fluid flow problems in micro-fluidic devices. A central feature in these applications is the formation of a thin ($\sim$ 10nm) and charged layer known as the Electric Double Layer (EDL) near charged surfaces. Numerical resolution of EDL is challenging and computationally expensive since physical quantities, such as electric potential and ion concentration, vary exponentially across this layer.

To address this challenge, we present numerical methods based on adaptive Quadtree grids (two spatial dimensions) and Octree grids (three spatial dimensions) that enable accurate, yet efficient, calculation of quantities of interest inside the EDL. Specifically, we present algorithms for the solution of Poisson-Boltzmann (PB) and Poisson-Nernst-Planck (PNP) Partial Differential Equations (PDEs).

These governing equations are widely used to study these various electrochemical systems as mentioned above.

Finally, we present several examples which illustrate the convergence of presented algorithms and provide two applications, namely in studying the potential distribution around charged biomolecules and in studying the charging dynamics of supercapacitors. Interestingly, in the second application, our detailed calculations provide unreported insights into the role of pore micro-structure on the charging kinetics of porous electrode. Specifically, we find that certain micro-structures that provide continuous conducting pathways (e.g. regular arrays of carbon nanotubes (CNTs) or graphene sheets) allow for high-density surface currents which decrease the charging times and hence can potentially improve the power density of porous electrodes.

Professor Frederic Gibou
Dissertation Committee Chair

Contents

Chapter 1

Introduction

Electrochemistry is concerned with the study of connection between electrical and chemical effects in charged solutions such as aqueous electrolytes and ionic liquids [106, 11]. Perhaps the most familiar example is a battery which converts chemical energy into electrical energy. Other examples could include technologies such as electroplating, fuel cells, and energy storage systems like batteries and supercapacitors [106, 45], or electrokinetics phenomena such as electrophoresis and electro-osmosis [93, 129]. Although the details, e.g. the chemical reactions that could happen at the surfaces, could be different among these systems, the fundamental transport mechanisms studied in this dissertation are quite universal and can be applied to many of electrochemical systems.

At the macroscopic level, an aqueous electrolyte may be represented by a set of concentration fields representing the number density each of individual ion species. Since ions are electrically charged, they interact with each other, charged surfaces (e.g. an electrode), or with any possible background electric field. At the continuum level, these individual interactions may be modeled via interactions with a mean electric potential field. Thus, at the simplest level, two physical mechanism that are responsible for the transport of ions in electrolytes are the familiar diffusion and motion of ions under electric force or “electro-migration”. Of course, this simple picture ignores other forms of transport that may be present. For instance, ions could be advected via a background fluid velocity field such as those set up by electro-osmotic or pressure driven flows. Moreover, this simple picture is only applicable to dilute electrolytes at small potentials and does not include steric [72, 14] or ion-ion correlation effects [15] that become important for concentrated solutions and at higher potentials [61].

Nonetheless, this simple model can be used to explain, or at least provide insight into, many different electrochemical systems. Importantly, it can be used to explain the so-called Electric Double Layer (EDL) that is formed at the interface of electrolyte with charged surfaces. Consider a simple setup consisting of an electrolyte made up two monovalent ion species (e.g. aqueous KCl) in contact with a charged surface as illustrated in figure 1. Initially the electric field due to the charge on the drives counter-ions (cations here) into the EDL and co-ions (anions here) out of the EDL. This results in the formation of charge cloud inside the EDL with an opposite sign to that on wall. As a result, the strength of the electric field is reduced due to partial screening of EDL. Moreover, accumulation of counter-ions and depletion of co-ions inside the EDL creates an imbalance in the concentration fields relative to the "bulk" of the electrolyte, far from the charged surface. As result, a diffusive flux is set up which acts against the electric field to "smooth out" the imbalance in the concentration field and further reduces number of ions that enter or exit the EDL.

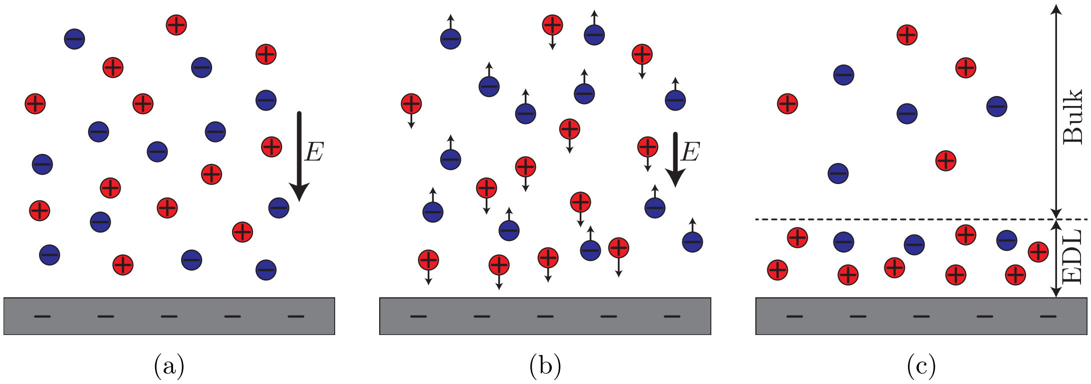


Figure 1.1: Formation of Electric Double Layer (EDL) around charged surfaces. (a) Initially concentration of both ions is the same everywhere and system is electro-neutral. (b) The electric field due to the charges on the surface exerts electric field on the cations and anions, deriving them in different directions. This rearrangement of ions gradually forms a layer with a net positive charge near the surface which is known as the Electric Double Layer (EDL). As the EDL charges up, its positive charge *screens* the negative charge on the wall, reducing the strength of electric field and thus slowing the charging process. (c) At equilibrium, the charge on wall is completely screened by an opposite charge in the EDL. The balance between the diffusive force, which tends to smear out gradients in the concentration, and the electric force, which tends to polarize the electrolyte, determines the thickness of the EDL. (c.f. equation (1.1))

Eventually, diffusion and electro-migration forces balance each other and the system reaches a thermodynamic equilibrium. At equilibrium, the thickness of

EDL, for a binary, symmetric electrolyte may be expressed as [129]:

$$\lambda_D = \sqrt{\frac{\varepsilon_0 \varepsilon_r k_{\mathrm{B}} T}{2 c_\infty e^2 z^2}}. \tag{1.1}$$

In this equation, $\varepsilon_0 = 8.85 \times 10^{-12}\,\mathrm{F\ m^{-1}}$ is the electric permittivity of vacuum, ε_r is the relative permittivity of electrolyte and for aqueous electrolytes is typically that of water, e.g. $\varepsilon_r \approx 78.2$ at room temperature, $k_{\mathrm{B}} = 1.38 \times 10^{-23}$ J $\mathrm{K^{-1}}$ is the Boltzmann constant, T is the temperature in Kelvin, z is the ion valency (e.g. $z = 1$ for $\mathrm{K^+}$ and $z = -1$ for $\mathrm{Cl^-}$ in aqueous KCl), $e = 1.6 \times 10^{-19}$C is the elementary charge of a proton and c_∞ is the number density of electrolyte in the bulk in terms of numbers per volume. At room temperature, i.e. $T = 298\,\mathrm{K}$, and for a milli-molar KCl, i.e. $c_\infty = 1\,\mathrm{mM}$, the Debye length is about 10 nm which is considerably thicker than the typical size of ions which are in the range of 0.2 to 2 Å [28].

It is possible to show that at equilibrium, ion concentrations inside the EDL follow the Boltzmann distribution [129]:

$$c_\pm = c_\infty \exp\left(\mp \frac{\psi e z}{k_{\mathrm{B}} T}\right),$$

where ψ is the electric potential inside the EDL relative to the bulk. The electric potential itself can be determined self-consistently by using the Poisson's equation:

$$\frac{\partial^2 \psi}{\partial y^2} = -\frac{\rho}{\varepsilon},$$

where $\varepsilon = \varepsilon_0 \varepsilon_r$, and $\rho = ez(c_+ - c_-) = -2ezc_\infty \sinh(\psi ez / k_\mathrm{B} T)$ is the free charge density inside the EDL, yielding,

$$\frac{\partial^2 \psi}{\partial y^2} = 2ezc_\infty \sinh\left(\frac{\psi ez}{k_\mathrm{B} T}\right). \tag{1.2}$$

Equation (1.2) is the one-dimensional Poisson-Boltzmann (PB) equation which describes the distribution of the electric potential inside the EDL. Assuming that the electric potential on the wall is fixed to a known value, i.e. $\psi(0) = v$, the PB equation (1.2) can be solved to yield:

$$\psi(y) = 4 \tanh^{-1}\left(\tanh\left(\frac{vez}{4k_\mathrm{B} T}\right) \exp\left(-\frac{y}{\lambda_D}\right)\right).$$

This is the so-called Gouy-Chapmann solution [129] which describes the equilibrium potential distribution inside the EDL. If the potential on the wall is small, i.e. $v/v_T \ll 1$ where $v_T = k_\mathrm{B} T / ez$ is the thermal potential, this equation further simples to the Debye-Huckel solution:

$$\psi(y) = v \exp\left(-\frac{y}{\lambda_D}\right). \tag{1.3}$$

which clearly demonstrates that the electric potential decays exponentially from its value on the wall to zero outside the EDL. It is also easy to note the role of the EDL thickness, λ_D, as the relevant length-scale.

The PB equation can easily be extended to two and three spacial dimensions as well (c.f. chapter 2). At higher dimensions, and especially for non-trivial boundaries, numerical solution of the equations are the only viable option. The exponential decay of potential inside the EDL, however, poses serious numerical difficulties especially if the macroscopic sizes of the domain are vastly different than that of EDL. In chapter 2 we will describe a class of numerical methods that utilize adaptive grids to overcome this difficulty without using excessive computational resources. A direct application of this algorithm is described in chapter 3 in calculation of electrostatic potential distribution on biomolecules. This is done by assuming the molecule as a dielectric body whose surface is calculated based on its atomic structure. A modified form of the PB is used to describe the potential distribution around biomolecules. This is because these molecules include discrete partial charges on separate atoms that need to be taken into account. Moreover, due to discontinuity of dielectric constant at the molecular boundary, appropriate boundary conditions must be implemented in form of jump conditions in the electric potential and electric displacement fields.

The main focus in this dissertation, however, is the subject of porous supercapacitors and the development of computational tools for studying such systems. The basic idea of storing energy in capacitors can be traced to very early days of developing a theory for electricity in the mid and late eighteenth century. It was not until one and half century later, however, that the works of Michael Faraday, and later on J. J. Thomson and Millikan on electron, led to developments in the theory of electricity at the molecular level. As a result, a true physical understanding of charging and discharging processes, as we now attribute to a flow of electrons in conducting medium, was not available until some 140 years later; although Faraday had, apparently, the first physically correct ideas on the polarization of dielectric materials in capacitors and their effect on the charge stored in them [45].

Instead of using a polarizable dielectric material, it is possible to think of capacitors that utilize electrolyte and rely on the formation EDL to store energy. Although the basic idea of making and using such devices has been known for years, the first patent dates back 1957 by Becker where a porous carbon electrode in an aqueous electrolyte was used to yield high surface area and thus maximize the stored charge and energy [17, 76]. Sohio Corporation [24], used the same ideas but incorporated non-aqueous electrolyte which enables to increase the working voltage to $3.4 - 4$V (as compared to $\sim$ 1V for aqueous electrolytes) and thus increase the amount of energy stored according to $U = 1/2\, CV^2$ [45, 75]. It was not until in the nineties, however, that supercapacitors became popular, mainly, due to their application in Hybrid Electric Vehicles (HEV) through U.S. DOE "ultracapacitor" program [76, 53, 128, 145].

The importance of the supercapacitors become more clear when compared against customary power sources like simple capacitors, batteries and fuel cells using the so-called Ragone plots [40, 75, 136]. Figure 1.2 is an example of a Ragone plot. In this figure, the vertical and horizontal axes represent the specific power (in W kg^{-1}) and specific energy (in Wh kg^{-1}), receptively. Devices with high specific power have the ability to deliver considerable amount of power while those with high specific energy storage can deliver power for longer times. As seen in this figure, normal capacitors rank higher than other devices with specific powers of order $\mathcal{O}(10^5)$ W kg^{-1} while lithium powered batteries rank almost three to four order of magnitudes lower. In contrast, these batteries have the potential to deliver power for a longer period, i.e. their specific energy are of the order $\mathcal{O}(10^2 - 10^3)$ Wh kg^{-1} as compared to capacitors which typically discharge in milliseconds. Interestingly, supercapacitors cover much wider spectrum in terms of both power and energy and are considered to be good candidates to improve power or energy ratings of power sources when used with respective devices. Furthermore, supercapacitors generally have longer cycle lives, $\mathcal{O}(10^5)$, than normal batteries, $\mathcal{O}(10^3)$ at best, as limited reactions happen at the electrode (if any) [45, 136].

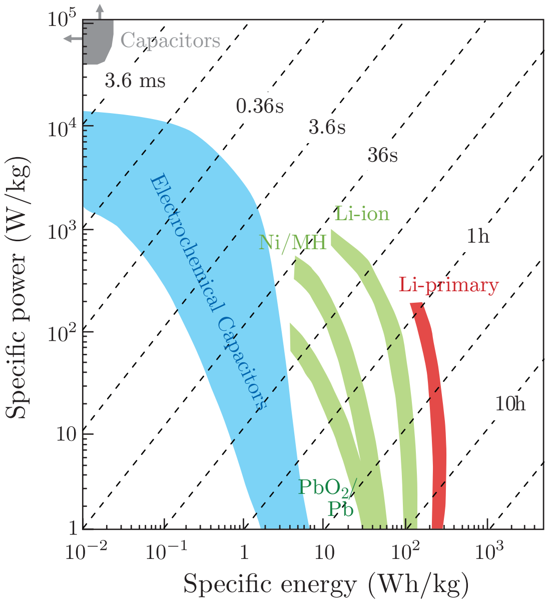


Figure 1.2: Ragone plot for typical power sources. Supercapacitros fill the gap between regular capacitors and batteries. (adopted from of Simon & Gogosti [136])

Generally, supercapacitors are classified into two group; those that are mainly based on charge storage inside the EDL and those that also involve oxidation-reduction (redox) processes [45, 136]. In the first group, charge and energy storage in the device is mainly due to polarization of electrolyte and formation of EDL as a response to external electric field. Using equation (1.3), the charge density on the wall may be calculated as:

$$q = -\varepsilon \frac{\partial \psi}{\partial y}|_{y=0} = \frac{\varepsilon}{\lambda_D} v,$$

which resembles the charge density stored on a parallel-plate capacitor whose capacitance is given by:

$$C = \frac{\varepsilon}{\lambda_D}, \tag{1.4}$$

with λ_D playing the role of separation distance between parallel plates. Using the value of $\varepsilon \approx 7 \times 10^{-10}$ F/m for water and $\lambda_D = 10$ nm yields a very high value of $C \approx 7\mu\text{F/cm}^2$. The small dimensions of EDL also allows them to form in very small pores and follow surface curvature. This means that porous electrodes which provide very high surface areas are perfect material candidates for supercapacitors, combining the high capacitance density of EDL with the high specific surface area of porous materials to boost the overall capacitance and energy storage.

Aside from large surface area, it is also necessary to use materials that are chemically stable, have good electrical conductivity, and are cheap to produce. Table 1.1 is a collection of some of the proposed materials in the literature. Carbon is by far the most popular option for the electrode material, especially in the form of activated carbon due to ease and low cost of its production. Other forms of carbon, such as aerogels, carbon nanotubes (CNT), and graphene sheets have also been suggested as possible materials. Interestingly, although electrodes based on activated carbon tend to produce larger surface area, it seems that CNT based electrodes seem to produce higher capacitances. This is usually justified by the fact that porous graphite based electrodes have wider spectrum of pore diameters and that electric double layer may not be formed at the micropores (radius < 2 nm). As a result the total *accessible* surface to the electrolyte is lower than CNT based electrodes where controlled synthesis of nano-tubes allows for more uniformly spaced pores that are easily accessible by the ions [111]. This theory, however, has recently been contraindicated by the works of [154] who report values as high as $C = 100-140$ F/g and $S = 1000-1600$ m^2/g for pores with an average diameter of $D_p = 7-11$ Å.

It is possible to boost the functionality of supercapacitors by incorporating materials that initiate surface redox reactions. This phenomenon, known as pseudo-capacitance, typically happens when electrode surface is covered with certain metal oxides such as RuO_2, Fe_3O_4, MnO_2, IrO_2, NiO or electronically conductive polymers [132, 76, 136]. In the case of Ruthenium oxide, RuO_2, for example this pseudo-capacitive behavior is attributed to Faradic reactions at the surface according to

$$\mathrm{RuO_2} + x\mathrm{H}^+ + x\mathrm{e}^- \rightleftarrows \mathrm{RuO}_{2-x}(\mathrm{OH})_x. \tag{1.5}$$

In pseudo-capacitance, energy storage is mainly achieved through the voltage-dependent proton absorption at the surface. This dependency may be modeled like a pseudo-capacitance by defining $C' = \partial(\Delta q)/\partial V$ where Δq is the amount of charge adsorbed at a specific electrode potential V. This type of supercapacitors have apparently first seen in the works of B. E. Conway [44] with metal oxides and typically result in much higher, sometimes $\mathcal{O}(10-100)$ times higher [46], specific capacitance than EDL based supercapcitors (c.f. table 1.1).

Although redox-based supercapacitors tend to achieve higher capacitance, they suffer from two main disadvantages compared to EDL based systems. First, they require considerably more expensive materials, e.g. the price for RuO_2 is \$ 1 per gram while carbon may be produced at the compelling price of only \$ 0.02 g^{-1} [157]. Furthermore, the chemical mechanism for these systems are similar to those involved in batteries and as a result they have fewer cycle lives than purely EDL based systems. Finally, progress has been made in combining both EDL and redox processes to benefit from their individual nice properties. For instance, extremely high capacitances have been in reported for hybrid systems that incorporate CNTs as beds for metal oxide particles or utilize nano-crystalline vanadium nitride (VN) for the electrode (c.f. table 1.1).

Electrochemical supercapacitors hold a promise for future use for different kinds of applications. Supercapacitors are especially beneficial to applications

C (F/g)	S (m^2/g)	P (kW/kg)	E (Wh/kg)	Material	Type	Ref.
500	1000 – 1500	–	–	activated carbon	EDL	[132]
15 – 16	400 – 1000	–	–	carbon foam	EDL	[132]
27.9 – 100	1150 – 2371	–	–	activated carbon	EDL	[58]
200 – 320	1900 – 3200	–	–	activated carbon	EDL	[58]
99 – 135	970	0.5 – 0.9	5 – 8.5	graphene sheets	EDL	[141]
45 – 95	400 – 800	–	–	carbon aerogel	EDL	[147, 119]
70 – 150	500 – 900	–	–	carbon aerogel	EDL	[58]
13 – 113	430	8 – 10	0.56	carbon nanotubes	EDL	[111]
180	375	20	7	carbon nanotubes	EDL	[81]
113	430	–	–	carbon nanotubes	EDL	[58]
100 – 140	1000 – 1600	–	–	carbon nanotubes	EDL	[154]
720	–	–	26.7	RuO_2	redox	[157]
634 – 768	–	–	26	RuO_2	redox	[158]
150	–	–	–	MnO_2	redox	[136]
930 – 1380	–	–	–	MnO_2	redox	[146]
200	–	–	–	MnO_2	redox	[80]
59	–	–	–	NiO	redox	[140]
50 – 64	–	11 – 17	7.9 – 11.1	NiO	redox	[86]
–	–	–	11 – 39	conducting polymer	redox	[128]
–	–	10.2 – 30.5	25 – 46	conducting polymer	redox	[105]
1300	–	–	–	VN	hybrid	[39]
1210	–	–	–	RuO_2 + conducting polymer	hybrid	[94]
930 – 1380	–	–	–	MnO_2 + platinum	hybrid	[146]

Table 1.1: Various types of supercapacitors in the literature along with the reported specific capacitance (C), specific surface area (S), specific power (P), and specific energy (E). Supercapacitors are largely classified into two groups; those that mainly depend on the electrolyte polarization and formation of EDL to store charge and those in which redox processes, such as (1.5), are the main mechanism for energy storage. Recently hybrid types have been proposed that combine the benefits of both types in a single device.

that require high power density and/or long durability. They are of considerable interest for energy recovery systems, for example, as in braking systems for vehicles and industrial machines [136] or during load lowering in large cranes [98]. Moreover, they may be used for other applications like power backup in video recorders, car audio systems, alarm clock radios, mobile phones and pagers, to name a few [76]. Other applications are in devices where large instantaneous currents are needed over a very short period of time, i.e. $\mathcal{O}\,(\mathrm{ms}-\mathrm{s})$, as in some toys or starters in Diesel locomotives [76]. Another possible application are in the fail safe systems that require instantaneous opening or closing operations, e.g. ECs are now being used in the new Airbus A380 to provide the high power needed for opening mechanism of emergency doors [136]. The transportation industry, however, still seems to be the main near future target market for these types of power sources [76, 136, 98].

Studying of transport processes in supercapacitors is crucial for understanding charging and discharging kinetics of these devices. As a result they are directly related to identifying important parameters that can boost the power density and reduce the charging times. Studying these systems are also beneficial to other technologies that utilize porous electrodes such as electro-catalysts [11], batteries [107], and fuel cells [123] as well as in emerging technologies like capacitive deionization (CDI) [143, 121, 120, 144], and 'blue' energy harvesting through capacitive mixing of fresh and salt water (CAPMIX) [26, 126, 125].

Theoretical studies of supercapacitors has largely been limited to simple circuit models such the infamous Transmission Line (TL) model [51]. Although the TL model can provide valuable insight into the charging dynamics and relevant time scales, it is technically a linear model that is only applicable if the EDL is thin compared to the pore radius and the applied potential is small (around 25 mV). Both of these assumptions, however, usually break down in most practical systems [136]. Importantly, at applied potentials larger than one thermal volt, EDL capacitance increases, creating a depletion zone in the bulk of electrolyte that is mostly devoid of ions. As detailed in chapter 5, this increases the resistance to transport of ions and thus reduces the power density of the electrode. In [20], authors provide modifications to the TL model to include these nonlinear effects at higher potentials.

Nonetheless, both TL and its nonlinear modification fail to appreciate the importance of electrode micro-structure in the charging kinetics. Indeed this is one of the most important aspects and contributions of this dissertation and is detailed in chapter 5. This is accomplished by performing Direct Numerical Simulations of Poisson-Nernst-Planck (PNP) equations (c.f. chapter 4) at the pore level and comparing certain metrics to those that are also computable through circuit models. As we shall see in chapter 5, our detailed numerical calculation reveals interesting, and unreported, physical phenomena that can accelerate the charging kinetics and thus potentially increase the power density for certain pore micro-structures.

With this introduction, we shall cover the following topics in the following chapters. First, in the next chapter we will review the Poisson-Boltzmann equation and detail a numerical algorithm for its discretization on adaptive Cartesian grids. This algorithm is then further modified to be suitable for bimolecular calculations in chapter 3. Next, in chapter 4 we present a conservative and novel hybrid finite difference/finite volume discretization for the PNP equations and study its convergence properties. Finally, chapter 5 presents the application of our PNP solver to study the charging kinetics of porous electrodes where we identify the surface conduction phenomena as a short-circuit mechanism that can accelerate the charging kinetics and, potentially, boost the power density of porous supercapacitors.

Chapter 2

Discretization of Poisson-Boltzmann Equation

2.1 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 applications, from colloid science and micro-fluidics [129, 93, 150] to biochemistry and biophysics [96, 134]. 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 [129, 93]. 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. [63], [50], [109] and [91] 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 [2] , [70], [92] and [150]. 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 [68, 47, 48, 66, 7]. 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 adaptive grids which would even further increase the computational efficiency. One major drawback, however, is that since elements need to conform to the boundaries and elements 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 [151, 71, 124]. 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 [67].

In this chapter 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 2.2), 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.

2.2 Poisson-Boltzmann equation

Consider a particle or a surface that has a fixed surface electrostatic potential due to surface charges. As explaind in chapter 1,when such a body is immersed inside an electrolyte, the electric field attracts the counter-ions and repels the co-ions in the solution to form the Electric Double Layer (EDL). Inside the EDL, and at equilibrium, ion densities follow the Boltzmann distribution [129, 93], i.e

$$c_i = c_\infty \exp\left(\frac{-z_i e}{k_\mathrm{B} T}\psi\right),$$

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:

$$\rho_e = ze\left(c_+ - c_-\right) = -2zec_\infty \sinh\left(\frac{ze}{k_\mathrm{B} T}\psi\right).$$

The Poisson-Boltzmann (PB) equation is obtained by using the expression for the charge density in the Poisson equation for electrostatic potential field,

$$\nabla^2 \psi = \frac{-\rho_e}{\varepsilon} = \frac{2zec_\infty}{\varepsilon} \sinh\left(\frac{ze}{k_\mathrm{B} T}\psi\right).$$

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

$$\lambda_D = \sqrt{\frac{\varepsilon k_{\mathrm{B}} T}{2 c_\infty z^2 e^2}},$$

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

$$x_i = L\tilde{x}_i \qquad \nabla = L^{-1}\tilde{\nabla} \qquad \psi = \frac{k_{\mathrm{B}} T}{ze}\tilde{\psi} \qquad \kappa = \frac{L}{\lambda_D}$$

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. In [68], authors 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 work, however, we will be using an iterative scheme where the solution at the previous step is used to expand the nonlinear term about, i.e we write:

$$\nabla^2 \psi^{\nu+1} = \kappa^2 \left(\sinh\left(\psi^{\nu}\right) + \left(\psi^{\nu+1} - \psi^{\nu}\right)\cosh\left(\psi^{\nu}\right)\right) + \mathcal{O}\left(\left(\psi^{\nu+1} - \psi^{\nu}\right)^2\right). \quad (2.2)$$

2.3 Numerical method

2.3.1 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 $|\nabla \phi| = 1$. Note, however, that although 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 (cells) that are represented by an Octree (Quadtree) data structure in three (two) spatial dimensions. Figure 2.1 (left) illustrates a two-dimensional 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 [101] and [142], one such criterion may be chosen as to divide a cell, with vertices in the set V, if the following inequality holds true:

$$\min_{v\in V} |\phi\,(v)| < \frac{L\,D}{2}, \tag{2.3}$$

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

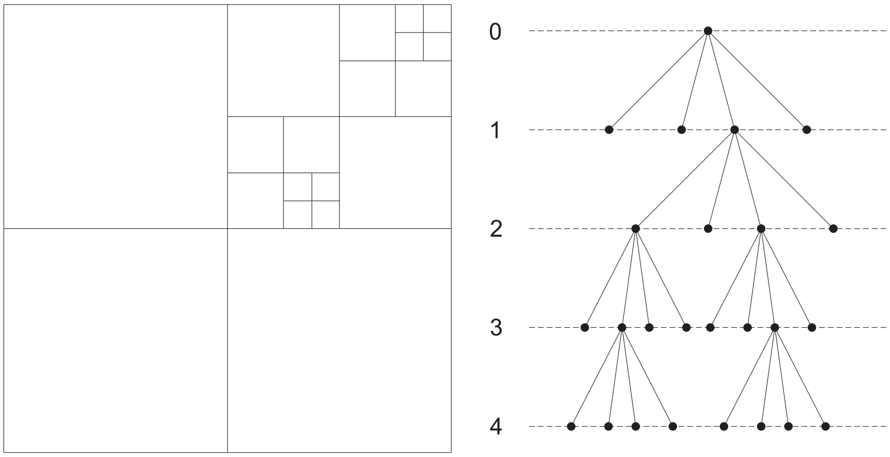


Figure 2.1: Non-graded adaptive discretization of a two-dimensional domain (left) and its corresponding Quadtree (right).

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 work we consider non-graded Cartesian grids where the solution is sampled at the nodes of each cell. Furthermore, 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 Figure 2.2)

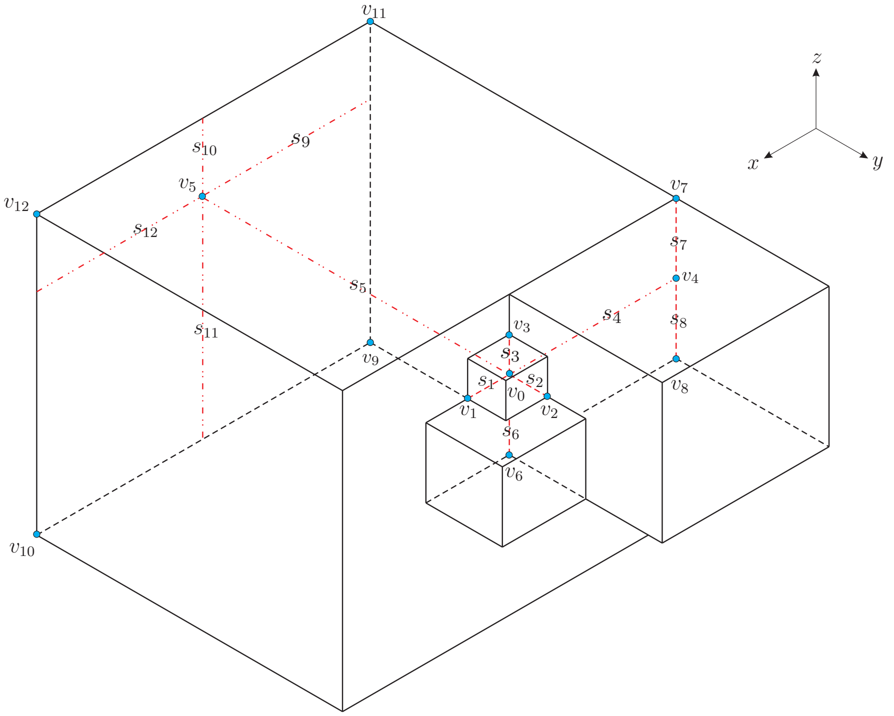


Figure 2.2: A general configuration for a three-dimensional mesh. Node v_0 has a three-dimensional T-junction in negative y direction (called v_5) and a two-dimensional T-junction in negative x direction (called v_4).

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,

$$\left\{ \begin{array}{rclcc} \left(\nabla^2 - \kappa^2 \cosh\left(\psi^n\right)\right)\psi^{n+1} & = & \kappa^2\left(\sinh\left(\psi^n\right) - \psi^n \cosh\left(\psi^n\right)\right) & \text{in} & \Omega^- \\ \psi & = & g\left(\vec{x}\right) & \text{on} & \Gamma \end{array} \right. . \tag{2.4}$$

To obtain a finite difference approximation to the Laplacian operator on the Octree mesh, we follow the works of [101] and [35]. Figure 2.2 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:

$$\psi_4^g = \frac{s_7\psi_8 + s_8\psi_7}{s_7 + s_8}, \tag{2.5}$$

and,

$$\psi_5^g = \frac{s_{11}s_{12}\psi_{11} + s_{11}s_9\psi_{12} + s_{10}s_{12}\psi_9 + s_{10}s_9\psi_{10}}{\left(s_9 + s_{12}\right)\left(s_{10} + s_{11}\right)}. \tag{2.6}$$

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. In [101] authors showed that it is always possible to eliminate the spurious error terms by appropriate weighting of Laplacian operator as:

$$\begin{aligned}
\nabla^2 \psi_0 &= \left(\frac{\psi_1 - \psi_0}{s_1} - \frac{\psi_0 - \psi_4^g}{s_4}\right) \frac{2\,\alpha}{s_4 + s_1} \\
&+ \left(\frac{\psi_2 - \psi_0}{s_2} - \frac{\psi_0 - \psi_5^g}{s_5}\right) \frac{2}{s_5 + s_2} \\
&+ \left(\frac{\psi_3 - \psi_0}{s_3} - \frac{\psi_0 - \psi_6}{s_6}\right) \frac{2\,\beta}{s_6 + s_3} + \mathcal{O}\left(h\right) \qquad (2.7)
\end{aligned}$$

where $h = \max\{s_i\}$, ψ_4^g and ψ_5^g are obtained through Eqs. (2.5) and (2.6), respectively, and α and β are given by,

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

Furthermore, they show that the finite difference matrix produced by Eq. (2.7) 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.

2.3.3 Treatment of the boundary condition

Equation (2.7) 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 chapter we will only address the Dirichlet boundary condition on the interface Γ and refer the interested reader to [65] for the implementation of Neumann and Robin boundary conditions for the PB equation.

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 figure 2.3 where the computational domain is to the left of the interface Γ. Note that condition (2.3) 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:

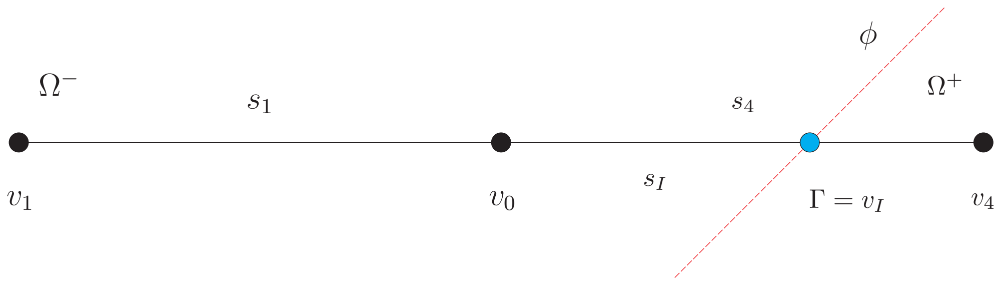


Figure 2.3: The interface intersects the domain between nodes v_0 and v_4 at $\Gamma = v_I$

$$\psi_{xx} = \left(\frac{\psi_I - \psi_0}{s_I} - \frac{\psi_0 - \psi_1}{s_1} \right) \frac{2}{s_I + s_1} + \mathcal{O}\left(h\right), \tag{2.8}$$

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

$$\phi\left(v_I\right) = \phi\left(v_0\right) + s_I\, \phi_x\left(v_0\right) + \frac{s_I^2}{2}\, \phi_{xx}\left(v_0\right) \tag{2.9}$$

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

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

Finally, by definition we know that $\phi\left(v_I\right)=0$ so that Eq. (2.9) may be solved for s_I as:

$$s_I=\begin{cases}\frac{-\phi_x(v_0)+\sqrt{\phi_x^2(v_0)-2\phi_{xx}(v_0)\,\phi(v_0)}}{\phi_{xx}(v_0)} & \phi_{xx}\left(v_0\right)>\epsilon,\\ -\frac{\phi(v_0)}{\phi_x(v_0)} & \left|\phi_{xx}\left(v_0\right)\right|\leq\epsilon.\end{cases}\tag{2.10}$$

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

2.3.4 Note on computing gradients near the interface

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 $\vec{E}=-\nabla\psi$, near the interface. In [6] authors proposed a PDE approach to extrapolate the solution outside the computational domain and [99] 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 $\psi_{nn}=\vec{n}\cdot\nabla\left(\vec{n}\cdot\nabla\psi\right)$ in Ω^-

and extrapolate it across the interface by solving:

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

where $H\left(\phi, \psi_{nn}\right)$ 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:

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

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

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

The Heaviside function $\left. H(\phi, V)\right|_{v_i}$ 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:

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

where $ngbd(v_i)$ denotes the set of direct neighboring nodes of the node v_i. One should refer to [99] 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 figure 2.4. Note the smooth extension of original solution (blue) in the Ω^+ region as shown by the red surface.

2.4 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.

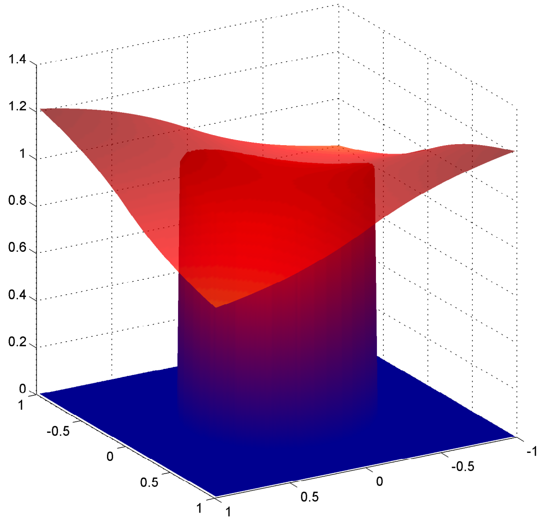


Figure 2.4: Extrapolation procedure: Blue and red surfaces show the solution before and after extrapolation, respectively. Note the smooth extension of solution in Ω^+ on the red surface

2.4.1 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:

$$\left\{ \begin{array}{rcl} \nabla^2 \psi & = & \sinh(\psi) + f_0, \\ f_0 & = & \nabla^2 \psi_0 - \sinh(\psi_0), \end{array} \right. \tag{2.11}$$

subject to a Dirichlet boundary condition given by the exact solution. Figure 2.5 illustrates the interface along with the adaptive grid.

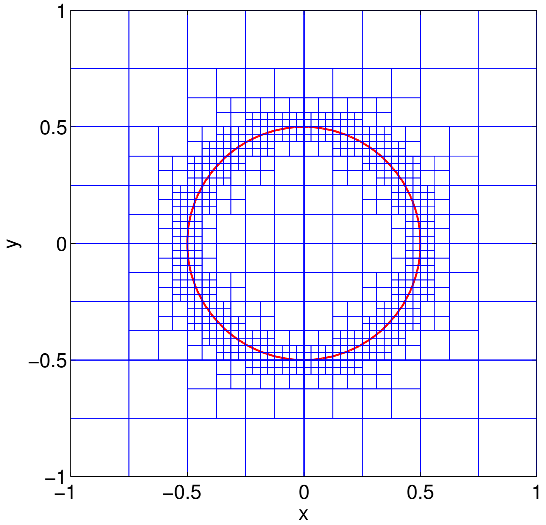


Figure 2.5: A circular interface along with the adaptive grid

Table 2.4.1 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.

$(res_{\min}, res_{\max})$	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)	29689	4.924×10^{-8}	2.718	4.588×10^{-7}	2.632

Table 2.1: Accuracy analysis for example 1

2.4.2 Example 2: A two dimensional spiky interface

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

$$\begin{aligned}\phi &= \min\left\{\phi_1, \phi_2\right\}, \\ \phi_1 &= \sqrt{|x| + 2\,|y|} - 0.9, \\ \phi_2 &= \sqrt{2\,|x| + |y|} - 0.9,\end{aligned}$$

with the same exact solution as in example 2.1. Figure 2.6 depicts the interface along with the computational grid. As seen in table 2.4.2, even when the interface is not smooth, our finite difference scheme produces results that are second-order accurate.

$(res_{\min}, res_{\max})$	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)	18897	2.190×10^{-7}	2.514	2.585×10^{-6}	2.352
(128,1024)	46581	3.511×10^{-8}	2.641	3.985×10^{-7}	2.064

Table 2.2: Accuracy analysis for example 2

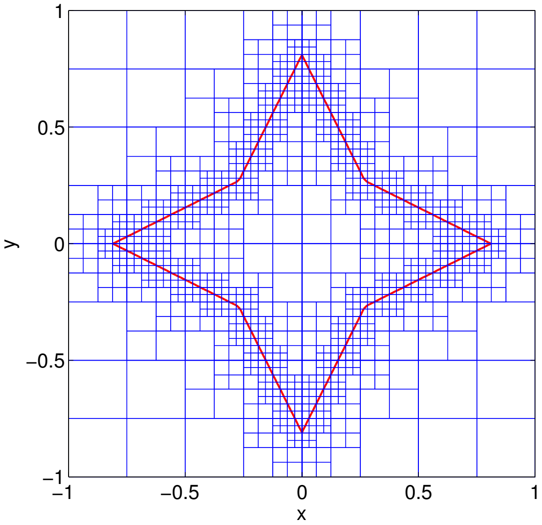


Figure 2.6: A two dimensional spiky interface along with the adaptive grid.

2.4.3 Example 3: An irregular interface

Finally consider an irregular interface as demonstrated in figure 2.7 with same exact solution as given in example 2.1. Table 2.4.3 illustrates the accuracy analysis of this example.

$(res_{\min}, res_{\max})$	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)	18897	2.241×10^{-7}	2.336	2.792×10^{-6}	2.352
(128,1024)	46581	3.488×10^{-8}	2.684	6.675×10^{-7}	2.064

Table 2.3: Accuracy analysis for example 3

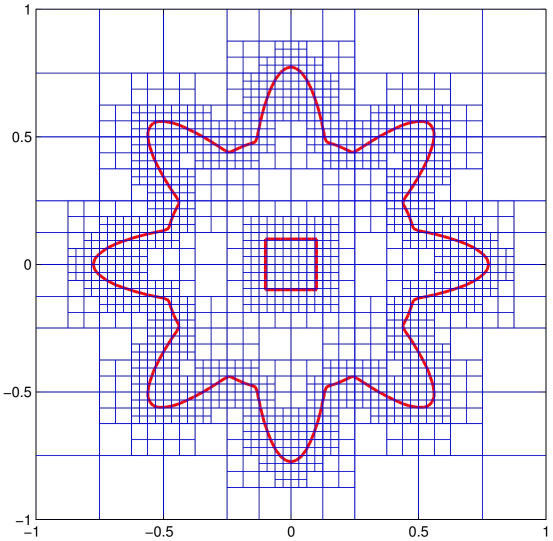


Figure 2.7: An irregular interface along with the adaptive grid.

2.5 Examples in three spatial dimensions

2.5.1 Example 1: Single particles

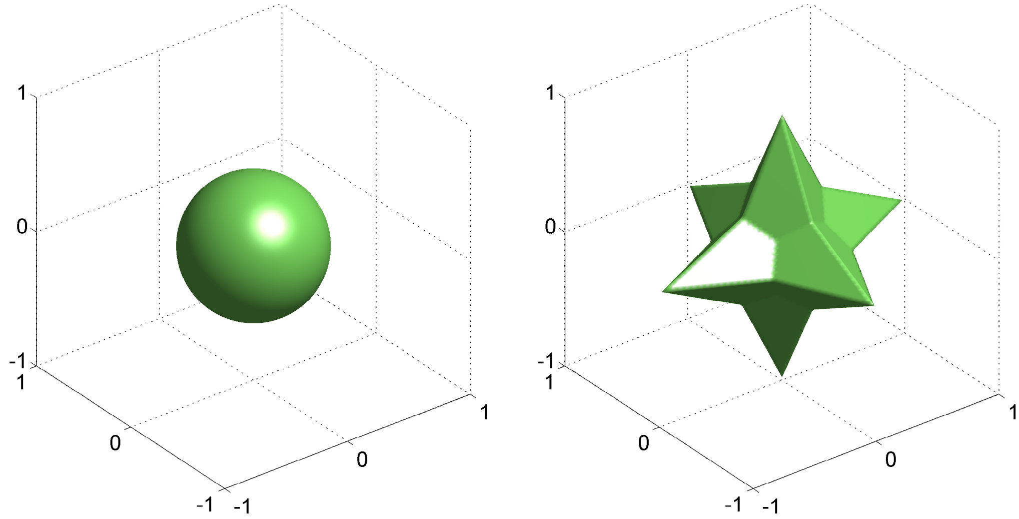
As for the first example let us consider the case where $\Omega = [-1, 1]^3$ where the exact solution is given by $\psi_0(x, y, z) = \sin(2\pi x)\ \sin(2\pi y)\ \sin(2\pi z)$. We are thus seeking a numerical approximation to the solution of Eq. (2.11) in three spatial dimensions subject to the Dirichlet boundary condition given by the exact solution. Two different types of interfaces are considered. Figure 2.8 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,

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

respectively.



(a) A Sphere with a radius of 0.5

(b) A three dimensional spiky interface

Figure 2.8: Interfaces used in the first example

Tables 2.5.1 and 2.5.1 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.

$(res_{\min}, res_{\max})$	Grid points	L_1 Error	Order	L_∞ Error	Order
(2,16)	2585	9.866×10^{-2}	-	9.612×10^{-1}	-
(4,32)	17889	1.500×10^{-3}	6.039	6.502×10^{-3}	7.207
(8,64)	132545	4.314×10^{-4}	1.798	1.751×10^{-3}	1.893
(16,128)	1019265	1.161×10^{-4}	1.894	4.605×10^{-4}	1.927

Table 2.4: Accuracy analysis for the sphere.

$(res_{\min}, res_{\max})$	Grid points	L_1 Error	Order	L_∞ Error	Order
(2,16)	3793	1.012×10^{-2}	-	2.911×10^{-1}	-
(4,32)	26257	2.408×10^{-4}	5.393	1.339×10^{-3}	7.764
(8,64)	194593	6.285×10^{-5}	1.938	3.272×10^{-4}	2.033
(16,128)	1496641	1.608×10^{-5}	1.966	8.071×10^{-5}	2.019

Table 2.5: Accuracy analysis for the rough three dimensional surface.

2.5.2 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 $\psi_0(x,y,z) = \sin(2\pi x)\sin(2\pi y)\sin(2\pi z)$. Figure 2.9 illustrates the interface cross-section and its corresponding adaptive grid. In accordance with our previous results, Table 2.5.2 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:

$$\begin{cases} \psi\left(\vec{x}\right) = 1 & \text{on} \quad \Gamma \\ \psi\left(\vec{x}\right) = 0 & \text{on} \quad \partial\,\Omega \end{cases}. \tag{2.12}$$

$(res_{\min}, res_{\max})$	Grid points	L_1 Error	Order	L_∞ Error	Order
(2,16)	3013	1.02×10^{-1}	-	1.14×10^{-0}	-
(4,32)	21169	1.25×10^{-3}	6.36	6.51×10^{-3}	7.45
(8,64)	158209	3.60×10^{-4}	1.79	1.75×10^{-3}	1.86
(16,128)	1222273	9.71×10^{-5}	1.89	4.60×10^{-4}	1.92

Table 2.6: Accuracy analysis for example 2

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 field, it is possible to get the electrostatic forces acting on particles by integrating the electric stress tensor around the objects [129, 93], i.e

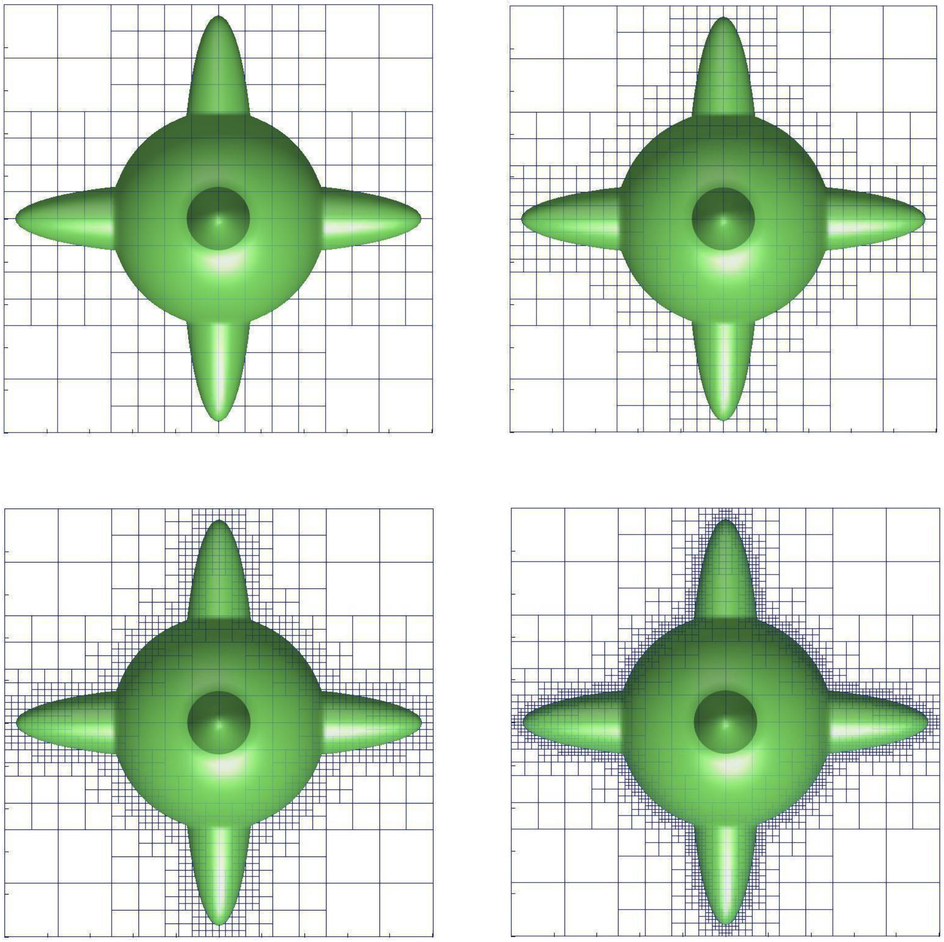


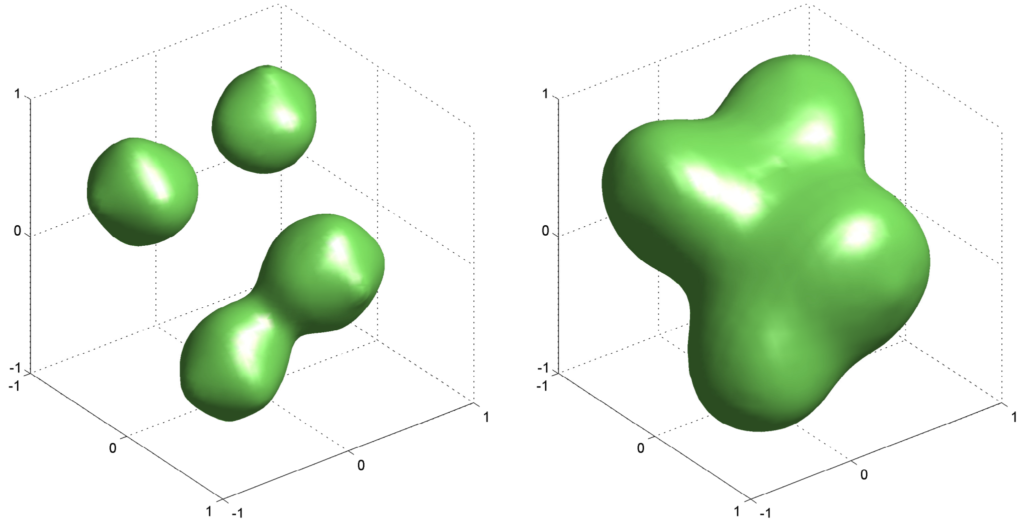
Figure 2.9: Adaptive grid generation as $res_{\max}$ is increased.

$$F_i = \int_S \sigma_{ij}\, n_j \mathrm{d}A \tag{2.13}$$

where σ_{ij} is given by,

$$\sigma_{ij} = -\Pi\delta_{ij} + \varepsilon\left(E_iE_j - \frac{1}{2}|E|^2\delta_{ij}\right), \tag{2.14}$$

where Π is the osmotic pressure and ε is the dielectric constant of the electrolyte. Solution results are shown in Figures 2.10 and 2.11. As indicated in section 2.3.4, 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 [100] for a reference on performing the surface integral in Eq. (2.13).



(a) $\psi = 0.6$

(b) $\psi = 0.2$

Figure 2.10: Isosurfaces of electrostatic potential ψ

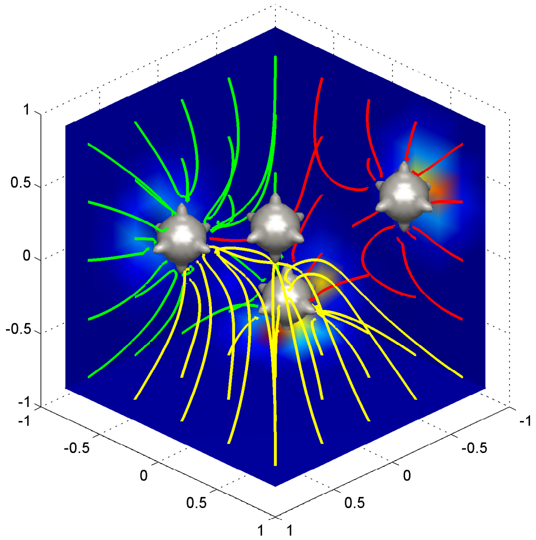


Figure 2.11: Electric fields. Each set of colored field line originates from a different face. Background faces are colored in term of the electric field magnitude.

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,

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

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. Figure 2.12 illustrates three different cross-sections of the interface along with the adaptive grid. Before considering the real boundary condition, and to analyze

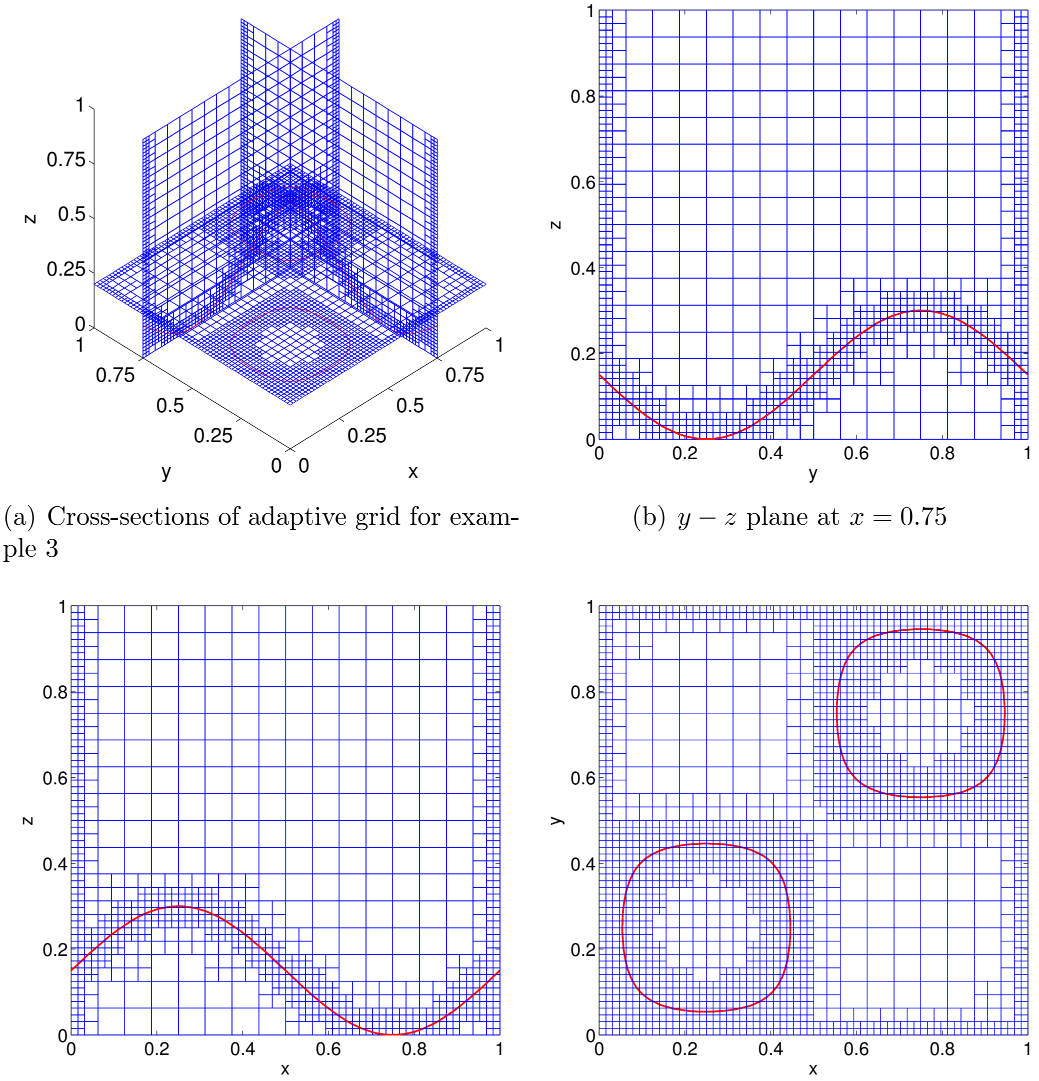
$(res_{\min}, res_{\max})$	Grid points	L_1 Error	Order	L_∞ Error	Order
(8,16)	3906	3.36×10^{-3}	-	1.01×10^{-2}	-
(16,32)	27107	7.37×10^{-4}	2.19	2.66×10^{-3}	1.92
(32,64)	200805	1.76×10^{-4}	2.07	7.06×10^{-4}	1.91

Table 2.7: Accuracy analysis for the roughness problem

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 2.5.3 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,

$$\begin{cases} \psi(x,y,z) = \zeta_0 & \text{on the surface} \\ \psi(x,0,z) = \psi(x,1,z) & \text{i.e periodic in x direction} \\ \psi(0,y,z) = \psi(1,y,z) & \text{i.e periodic in y direction} \\ \psi(x,y,z) \sim \alpha e^{-\kappa z} & z \to \infty \end{cases}, \tag{2.15}$$

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. (2.15). When applying this type of boundary condition numerically, the coefficient α is found by a simple iterative procedure.



(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 $z = 0.2$

Figure 2.12: Three dimensional adaptive grid and its cross-sections around the surface roughness

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. (2.16) and may be used to check the accuracy of our method. Table 2.5.3 also confirms our previous results in that our discretization is second-order accurate.

$$\psi = 4\,\tanh^{-1}\left(\tanh\left(\frac{\zeta_0}{4}\right)e^{-z}\right). \tag{2.16}$$

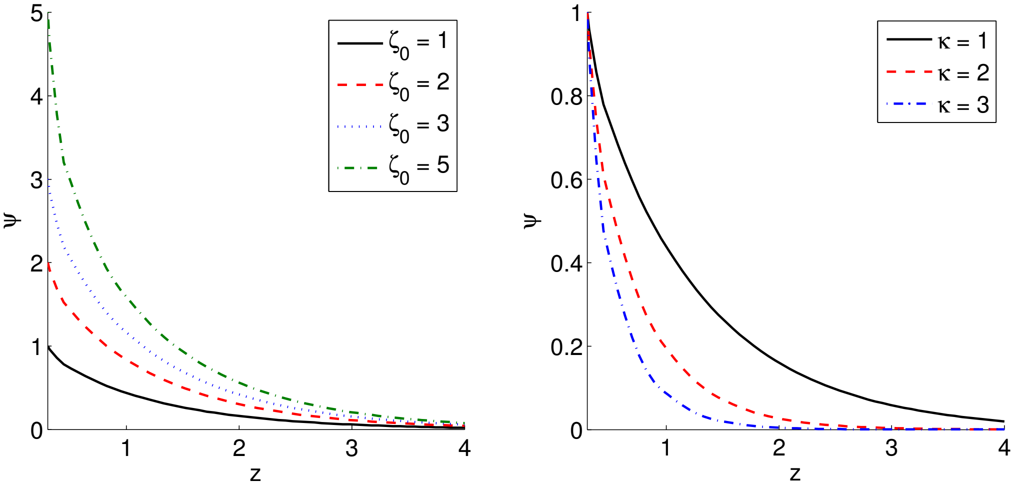
$(res_{\min}, res_{\max})$	Grid points	L_1 Error	Order	L_∞ Error	Order
(8,16)	3394	1.57×10^{-4}	-	4.00×10^{-4}	-
(16,32)	22819	3.91×10^{-5}	2.00	1.24×10^{-4}	1.69
(32,64)	165733	9.82×10^{-6}	1.99	3.28×10^{-5}	1.92

Table 2.8: Accuracy analysis for the roughness problem

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. 2.13(a) and 2.13(b) show the effect of these parameters on the potential drop across the EDL. The simulation was done on an adaptive grid with $(res_{\min}, res_{\max}) = (64, 16)$.

2.5.4 Remarks on the iteration scheme

Poisson-Boltzmann equation is non-linear and as discussed earlier we have used the Newton's iteration method to handle the non-linearity in an efficient way. It is



(a) $\kappa = 1$

(b) $\zeta_0 = 1$

Figure 2.13: Effects of surface potential and EDL thickness on the solution. Plots generated for $x = y = 0.25$ and $\delta = 0.15$

thus interesting to study the effect of various physical and numerical parameters, such as surface potential, Debye layer thickness and grid size, on the convergence of this iteration scheme. To do so, an absolute tolerance of 10^{-6} was chosen for the L_∞ norm of solution difference at two consecutive iterations, i.e we require that the following criterion to be met for the convergence

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

Two examples are considered: the sphere described in section 2.5.1 and the irregular surface described in section 2.5.2. In both cases, a range of surface potential $\zeta = 1 - 5$ was chosen for the boundary condition and the problem was solved both in the thick, $\kappa = 1$, and thin, $\kappa = 5$, double layer limits using a coarse, $(res_{\min}, res_{\max}) = (32, 128)$, and a fine, $(res_{\min}, res_{\max}) = (32, 512)$, computational grid. Table 2.5.4 and 2.5.4 illustrates the variations in the number of iterations needed for convergence for the sphere and irregular surface, respectively.

Both results suggest that the scheme converges very fast; with maximum the number of iterations needed for convergence being 6. Out of all various parameters, the iteration scheme depends mainly on the surface potential used as the boundary condition. This is expected since as the surface potential increases in magnitude, the non-linearity of the problem increases due to the $\sinh(\psi)$ term. Other parameters, however, do not play a strong role in the convergence properties of the iteration scheme method as they are not involved in the linearization approximation to the Poisson-Boltzmann equation.

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 2.9: Number of iterations needed for convergence for the sphere example (cf. section 2.5.1)

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

Table 2.10: Number of iterations needed for convergence for the irregular surface example (cf. section 2.5.2)

Chapter 3

Application of Poisson-Boltzmann Equation to Biomolecular Calculations

3.1 Introduction

The Poisson-Boltzmann equation is useful for calculating important biomolecular quantities such as binding energies [57]. However, solving this equation numerically has many challenges, the most significant of which are a) charge singularities, b) representing molecular surfaces, c) addressing exponential nonlinearities in the solution, and d) imposing the correct jump boundary condition. In this chapter, we extend the algorithm introduced in chapter 2 to address these computational challenges in novel ways. First, we describe a simple and robust technique for implicitly representing biomolecular surfaces. Next, we demonstrate a novel discretization method for imposing the correct jump boundary conditions on the surface. Finally, we validate the solver and show its usefulness by calculating solvation free energies.

Since the pioneering work of Warwicker and Watson in the early 1980s [148], many different techniques for solving the Poisson-Boltzmann equation have been developed, most of which are based on finite difference, finite element, or boundary element methods. Here, we do not intend to describe or compare them and refer the interested reader to [8, 9, 74, 36, 90] and the references therein for recent reviews.

An important characteristic of modern Poisson-Boltzmann solvers is the ability to use variable resolution. This allows to have coarse resolution where the solution is smooth and fine resolution where the solution varies rapidly. Indeed, one of the advantages of finite element methods over finite difference methods has been the robust adaptivity. Finite element methods are able to locally refine the computational mesh based on an error indicator, increasing resolution as needed, which enabled them to more efficiently address the exponential nonlinearity in the Poisson-Boltzmann equation [8, 90]. Finite difference solvers can achieve similar results through the practice of focusing, in which the equation is solved on a coarse mesh, and the solution is used as a boundary condition for a finer mesh over an interesting subdomain [62].

Recent works have introduced adaptive finite difference methods that discretize the Poisson-Boltzmann equation on non-uniform grids. In [25], Boschitsch and Fenley introduced a first-order method that solves the nonlinear Poisson-Boltzmann equation on a graded Octree mesh. In the previous chapter we presented a second-order method for solving the Poisson-Boltzmann equation on a non-graded Octree mesh. Neither approaches, however, use error estimates to refine the mesh. Instead, they refine the mesh based on distance from the molecular surface. The rational is based on the elliptic nature of the equation, which ensures that the solutions are smooth away from the interface.

Another difference between finite difference and finite element methods is that finite element methods ensure that cell edges align with interfaces. This is an appealing feature, as interfaces cutting through cells–as happens in finite difference schemes–complicate the discretization of boundary conditions. This property, however, comes at a price; creating a finite element mesh for a geometrically complicated domain, such as the surface of a protein, can be very expensive [37].

Finite difference methods, on the other hand, do not require the grid to conform to the boundary. As a result the grid generation, for uniform meshes, is trivial. However, since the grid does not conform to the boundary, special care must be taken to discretize the boundary conditions. This is specially important for biomolecular computations since one has to impose jump boundary conditions on complicated and, potentially, singular geometries. Still finite difference solvers are quite popular and important improvements have been done over the years. Notably, Wei and coworkers have developed a method, termed "Matched Interface and Boundary" (MIB), that is able to produce second-order accurate results through accurately imposing the jump boundary conditions. For more information on this method, one may refer to [59, 160].

The work presented here builds directly on that of chapter 2 with appropriate modifications to address charge singularities and jump boundary conditions on the biomolecular surface.

3.2 Domain description using level set functions

There are multiple ways of defining a molecular surface. The simplest approach is to use the *van der Waals Surface* (vdWS), which represents a molecule with a set of intersecting spheres of radii r_i, where r_i is the van der Waals radius of the i–th atom in the molecule. This surface is not completely accessible to solvent molecules, though, and therefore not appropriate for implicit solvent models. To address this, one can use the *Solvent Accessible Surface* (SAS), which is the set of spheres with radii $r_i + r_s$, where r_s is the solvent radius [79]. The SAS is commonly used to represent the hydration effects. Unfortunately, both vdWS and SAS result in geometrical singularities due to self intersection between spheres. To remedy this problem, it is possible to use the *Solvent Excluded Surface* (SES), which are the boundary points that are in contact with a solvent molecule as the solvent molecule "rolls" over the vdWS [127, 64]. These three surfaces are schematically depicted in figure 3.1.

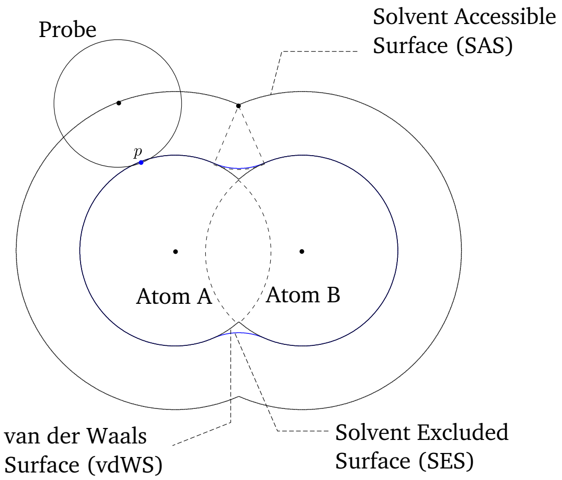


Figure 3.1: Common surfaces used in biomolecular computations. (Copyright Notice: First published in [103] by Global Science Press.)

Different methods have been proposed over the years to compute these surfaces. Connolly proposed an analytical algorithm for computing both the SAS and the SES [42, 43], and Sanner and Oslon described an algorithm for analytically computing the SES and providing a triangulated representation of the surface [131]. Analytic representation of the SES can lead to very accurate computation of molecular surface and volume. However, this technique requires geometric singularities to be dealt with explicitly. Alternatively, by using an implicit representation of the molecular surface, one can potentially avoid to handle any, or most of, singularities explicitly.

Level set methods, originally proposed by Osher and Sethian in [115], are a general, robust, and flexible framework for implicitly representing and tracking interfaces that undergo complex topological changes. Applying this idea to biomolecules, a molecular surface–vdWS, SAS, or SES–can be represented as the zero level set of a three dimensional function. More precisely, the level set function, $\phi(x, y, z)$, divides the domain, Ω, into two domains, Ω^+ and Ω^-, and the interface Γ,

$$\begin{cases} \Omega^+ & \equiv & \left\{\mathbf{x} \in \mathrm{R}^3 \mid \phi(x,y,z) > 0\right\} \\ \Gamma & \equiv & \left\{\mathbf{x} \in \mathrm{R}^3 \mid \phi(x,y,z) = 0\right\} \\ \Omega^- & \equiv & \left\{\mathbf{x} \in \mathrm{R}^3 \mid \phi(x,y,z) < 0\right\} \end{cases} .$$

A large body of work has focused on computing level set functions for various geometries. Here we describe the essentials, and we refer the reader to [133] and [114] for a more thorough survey of the methods. To generate a level set function, one normally starts with an initial function that correctly predicts the location of boundary. In the case of the vdWS surface, for example, this initial function may simply be chosen as

$$\phi_0(x, y, z) = \max_i \left\{ r_i - \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} \right\} .$$

However, it is not always possible to obtain a good initial function; for example, no such simple expression exists for the SES. Furthermore, to obtain a good adaptive grid and maintain robustness, it is required that the level set function be a distance function, i.e $|\nabla\phi| = 1$. To achieve this property, and once an initial level set function, ϕ_0, is chosen, the *reinitialization equation*,

$$\frac{\partial\phi}{\partial\tau} + S(\phi_0)(|\nabla\phi| - 1) = 0, \tag{3.1}$$

must be solved where τ is a fictitious time and S is the numerical sign function, usually taken as

$$S(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + \Delta x^2}}.$$

Compared to traditional methods, using a level set function to represent the molecular surface is a relatively new technique that has been explored by many different authors. In [32] the authors described a level set method for capturing the SES by starting from the vdWS and moving the interface in the normal direction twice to obtain the SAS and SES. At the end of second pass, however, the SES may contain inaccessible cavities. These are removed by a third pass in which a sphere is constructed around the outer SES and then “shrink-wrapped” to the molecule. Another approach obtains the SES by moving the boundary only once, but it must impose the correct curvature at toric segments [117]. Although defining the correct curvature seems reasonable, it leads to solving a nonlinear advection-diffusion equation describing the motion of level set, which is computationally expensive. Moreover, defining the correct curvature requires the definition of a reduced surface, leading to extra complication [131].

Alternatively, within the level set framework, it is relatively easy to "redefine" the molecular surface such that it minimizes the solvation free energy. Different approaches exists in the literature. For example, in [83] the authors obtain the molecular surface by evolving the level set function such that it minimizes a certain solvation free energy functional. In another article by Bates et al. [116], the authors use similar ideas to obtain the *Minimal Molecular Surface* (MMS) obtained via mean curvature minimizations.

The method presented here to generate the level set function is similar to [32] in that we do not explicitly enforce the curvature and aim for the classical definition of SES. However, instead of moving the level set in the normal direction twice, which would be expensive, we only reinitialize the level set and note the following:

1. The reinitialization equation, (3.1), is closely related to moving the interface in the normal direction,

2. After reinitialization, the level set $\phi = \pm d$ is at the distance $\pm d$ away from the interface along the normal direction, and

3. Toric segments are automatically generated by the rarefaction waves when unit normal vectors diverge (see figure 3.2).

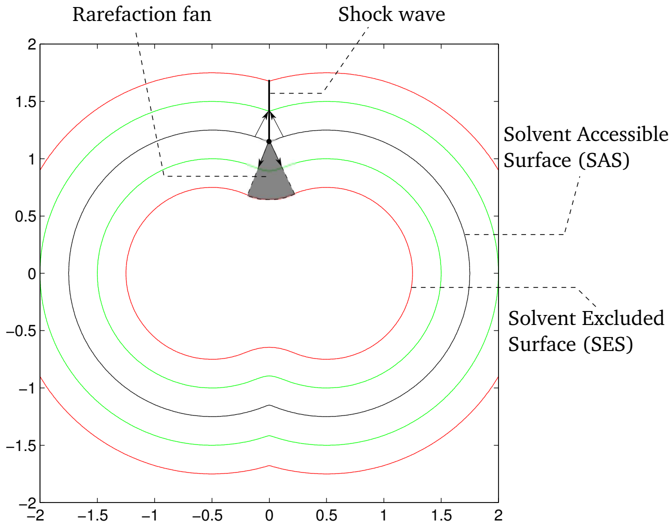


Figure 3.2: Schematic illustration of the SES generation algorithm. Once the SAS is reinitialized, rarefaction fan and shock waves propagate into the domain, depending on whether the normal vectors diverge or converge, respectively. This automatically ensures the formation of toric segments for all the level set contours in Ω^+, including the SES. (Copyright Notice: First published in [103] by Global Science Press.)

Within this new framework, generating the SES is quite simple and efficient. Our algorithm is the following:

1. Start with an approximation for the SAS by looping over all atoms in the molecule:

$$\phi_0(x, y, z) = \max_i \left\{ r_i + r_s - \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} \right\}. \quad (3.2)$$

2. Reinitialize the level set function ϕ' using equation (3.1) with ϕ_0 defined in equation (3.2) as the initial approximation.

3. Obtain the SES by taking the zero level set of $\phi = \phi' - r_s$.

Figure 3.2 schematically illustrates the application of this algorithm. In the rest of this chapter, we adopt the convention that Ω^+ refers to the inside of the molecule, Ω^- refers to the outside, and Γ represents the SES.

Finally we note that at the end of this algorithm, the SES may contain inner “cavities”. To identify and remove these cavities, we incorporate a simple fix. The basic idea is based on the observation that inner cavities are not, topologically, “connected” to the boundaries of the computational domain. As a result, any algorithm that can benefit from this observation, can detect the cavities and remove them. One such simple algorithm is to solve an auxiliary diffusion equation in the Ω^- domain subjected to a zero boundary condition on the SES surface and a nonzero boundary condition on the computational domain boundaries. Specifically, we solve:

$$\begin{aligned}
\nabla^2 \mathcal{C} &= 0, \qquad \mathbf{x} \in \Omega^-, \\
\mathcal{C}(\mathbf{x}) &= 0, \qquad \mathbf{x} \in \Gamma, \\
\mathcal{C}(\mathbf{x}) &= b, \qquad \mathbf{x} \in \partial\Omega,
\end{aligned}$$

where $\partial\Omega$ is the boundary of the computational domain, and b is a constant coefficient different from zero. It is clear that the solution of this problem, defined only in Ω^-, is equal to zero in the cavities, and is different from zero elsewhere. Once the solution is found, cavities are simply marked wherever $\phi < 0$ and $\mathcal{C} = 0$. The removal process simply consists of changing the sign of the level set function in the cavities. Note that a few iterations of the reinitialization equation may be required to avoid any discontinuity in the level set function due to changing the sign of the level set function inside the cavities. Figure 3.3 illustrates the application of this algorithm in removing the cavities inside the 2C00 molecule.

3.3 Note on octree grid generation

Refining the mesh locally is often preferred to keeping a uniform grid. The solution to the Poisson-Boltzmann equation is smooth away from the interface, but due to boundary conditions and formation of the electric double layer, could

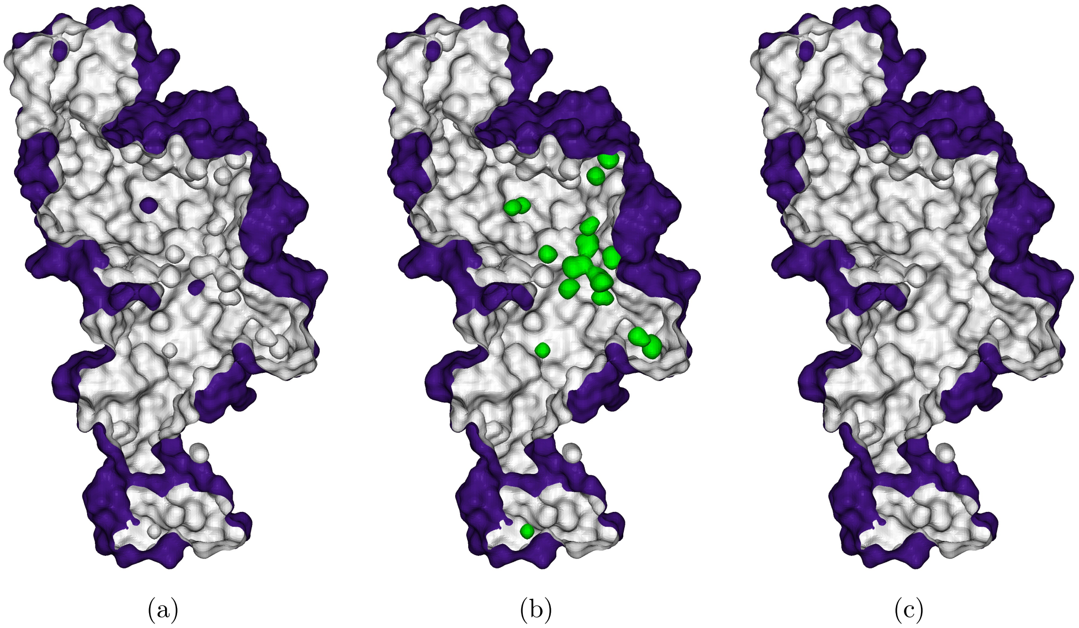


Figure 3.3: Cavity removal process. (a) A cross-section of the SES generated for the 2C00 molecule with outside colored in purple and inside in grey. (b) Cavities, colored in green, are detected inside the molecule by checking if $\phi < 0$ and $\mathcal{C} = 0$ at each grid point. (c) Cavities are removed by simply changing the sign of the level set function at the corresponding grid points. (Copyright Notice: First published in [103] by Global Science Press.)

have very large gradients near the interface. One way to address this problem is to introduce more grid cells near the interface. For three spatial dimensions, octree grids have been shown to be an optimal choice for local grid refinement [1].

When the refinement is performed near the interface, the number of grid points is proportional to the surface of the molecule rather than the volume of the computational domain. Moreover, for elliptic problems, the main factor determining the execution time and memory consumption is the size of the resulting linear system.

As such, an octree-based method can be many times faster and memory-efficient than a uniform method. The computational advantage of adaptive grids is particularly great when there are jumps in the solution at the interface, since errors in the jump or its location will propagate into the entire domain. Adaptive grids are also excellent at resolving rapidly changing solutions, such as the electrostatic potential inside the electric double layer. Adaptive grids are therefore advantageous for solving the Poisson-Boltzmann equation with singular terms. Here we follow the same technique described in the previous chapter for grid generation and thus refer the reader to section 2.3.1 for details.

3.4 Governing equations

The electrostatic potential, ψ, around a biomolecule in a symmetric, binary z:z electrolyte solution can be described by the Poisson-Boltzmann (PB) equation, modified to include the effects of singular charges on separate atoms:

$$-\nabla \cdot (\varepsilon \nabla \psi) + 2c_\infty(\mathbf{x}) ez \sinh(\psi) = \sum_{i=1}^{N_m} q_i \delta\left(\mathbf{x} - \mathbf{x}_i\right),$$

where ε is the permittivity of the electrolyte, e is the charge of a proton, z is the valence of the background electrolyte, c_∞ is the bulk salt concentration, k_B is the Boltzmann coefficient, T is absolute temperature, q_i is the atomic partial charge,

$\mathbf{x}_i$ is the location of each individual atom and N_m is the number of atoms in the molecule. Note that the explicit dependence of bulk salt concentration on position is only to indicate that mobile ions only exist in the solution; that is, $c_\infty(\mathbf{x}) = 0$ inside the molecule. Using the same non-dimensionalization technique described in section 2.2, we obtain:

$$-\nabla \cdot (\varepsilon \nabla \psi) + \kappa^2(\mathbf{x}) \sinh(\psi) = \sum_{i=1}^{N_m} z_i \delta\left(\mathbf{x} - \mathbf{x}_i\right), \tag{3.3}$$

where the potential has been scaled to the thermal voltage, electrolyte permittivity is scaled to that of vacuum, z_i is the non-dimensional partial charge on the atoms, and κ is the non-dimensional inverse of Debye length outside the molecule and $\kappa = 0$ inside. Equation (3.3) is accompanied with jump conditions at the molecular surface; that is, we require that

$$\begin{aligned} [\psi]_\Gamma &= 0, \\ [\varepsilon \nabla \psi \cdot \mathbf{n}]_\Gamma &= 0. \end{aligned} \tag{3.4}$$

3.4.1 Technique for representing singular charges

Finite difference methods typically use a Dirac delta function to map discrete charges onto the grid. This method can obtain second-order accuracy, but comes with challenges. First, if the mesh is too coarse, then charges near the molecular interface may actually be smeared out so much as to extend outside the molecular interface. Geng *et al.* found this effect to significantly reduce the accuracy of their second-order MIB solver [59]. Our adaptive meshing technique could address this by introducing additional grid points, but that would increase the computational cost.

Instead, as in Geng *et al.* [59], we choose to use a formulation that allows us to represent the singular charges through their effect on the molecular interface. This technique, introduced by Chern *et al.*[38], is to separate the singular part of the solution, associated with the discrete charges in the molecule, from the regular part. This introduces a modified jump condition at the interface, which differs from equation (3.4) as described below. First, the solution to the electrostatic potential, ψ, is split into regular, $\hat{\psi}$, and singular,$\bar{\psi}$, parts:

$$\psi = \hat{\psi} + \bar{\psi}.$$

The singular part of the potential is defined such that

$$\bar{\psi} = \begin{cases} \psi^* + \psi^0 & \text{if } \mathbf{x} \in \Omega^+, \\ 0 & \text{if } \mathbf{x} \in \Omega^-, \end{cases}$$

where ψ^* is the Coulombic potential due to singular charges,

$$\psi^* = \sum_{i=1}^{N_m} \frac{z_i}{4\pi\varepsilon^+} \frac{1}{|\mathbf{x} - \mathbf{x}_i|},$$

and ψ^0 fulfills:

$$\begin{aligned} \nabla^2 \psi^0 &= 0 \qquad \text{if } \mathbf{x} \in \Omega^+, \\ \psi^0 &= -\psi^* \qquad \text{on } \Gamma. \end{aligned}$$

Using this decomposition, the regular part of the solution may be obtained by solving

$$-\nabla \cdot \left(\varepsilon \nabla \hat{\psi} \right) + \kappa^2(\mathbf{x}) \sinh\left(\hat{\psi} \right) = 0, \tag{3.5}$$

subjected to the following modified jump conditions:

$$\begin{aligned} \left[\hat{\psi} \right]_\Gamma &= 0, \\ \left[\varepsilon \nabla \hat{\psi} \cdot \mathbf{n} \right]_\Gamma &= -\varepsilon^+ \nabla \left(\psi^* + \psi^0 \right) \cdot \mathbf{n}|_\Gamma. \end{aligned} \tag{3.6}$$

3.5 Spatial discretization

The main difficulty in deriving numerical methods for adaptive Cartesian meshes is addressing T-junctions accurately (see figure 2.2). This was explained in the previous chapter and we refer the reader to section 2.3.2 for the details. Away from the interface, this finite difference method is utilized to discretize equation

(3.5). Close to interfaces we use a finite volume approach to handle the jump conditions in equation (3.6). This approach is presented next. For clarity, we present the jump for the Poisson-Boltzmann equation in two spatial dimensions, but extending this technique to three spatial dimensions is straightforward.

3.5.1 Discretization near the interface

Consider the following general formulation of the Poisson-Boltzmann equation with variable coefficient, ε, and jump conditions:

$$\begin{aligned} -\nabla\cdot\left(\varepsilon\nabla\hat{\psi}\right)+\kappa^2(\mathbf{x})\sinh\left(\hat{\psi}\right) &= f\left(\mathbf{x}\right), \qquad \mathbf{x}\in\Omega, \\ \left[\hat{\psi}\right] &= a\left(\mathbf{x}\right), \qquad \mathbf{x}\in\Gamma, \\ \left[\varepsilon\nabla\hat{\psi}\cdot\mathbf{n}\right] &= b\left(\mathbf{x}\right), \qquad \mathbf{x}\in\Gamma, \end{aligned}$$

where $\hat{\psi}$ is a scalar to be solved for, $f\left(\mathbf{x}\right)$, $a\left(\mathbf{x}\right)$ and $b\left(\mathbf{x}\right)$ are known scalar functions, and $\left[\hat{\psi}\right]$ and $\left[\varepsilon\nabla\hat{\psi}\cdot\mathbf{n}\right]$ are the jump in the electrostatic potential, $\hat{\psi}$, and the normal component of the electric displacement field, $D=-\varepsilon\nabla\hat{\psi}$, respectively. As previously described, Ω is the whole domain that is split into two subdomains Ω^- and Ω^+ by the interface Γ. Variables are continuous inside each domain, but ε, $\hat{\psi}$ and f can be discontinuous across the interface. The grid configuration near the interface is depicted in figure 3.4. Integrating inside both Ω^- and Ω^+ over the

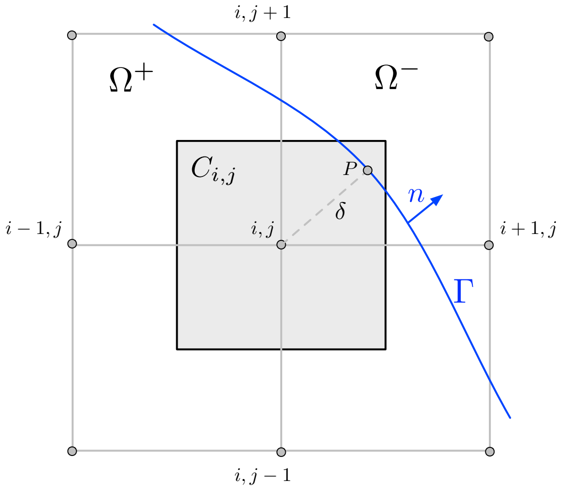


Figure 3.4: Cell $C_{i,j}$ surrounding the node located at i,j. The point P is the point on the interface Γ that is closest to the node. (Copyright Notice: First published in [103] by Global Science Press.)

cell surrounding node (i,j), denoted by $C_{i,j}$, leads to the following equations:

$$\begin{aligned}
-\int_{C_{i,j}\cap\Omega^-} \nabla\cdot\left(\varepsilon^-\nabla\hat{\psi}^-\right)\mathrm{d}\mathcal{A} + \int_{C_{i,j}\cap\Omega^-}\kappa^2(\mathbf{x})\sinh\left(\hat{\psi}^-\right)\mathrm{d}\mathcal{A} &= \int_{C_{i,j}\cap\Omega^-} f^-\mathrm{d}\mathcal{A},\\
-\int_{C_{i,j}\cap\Omega^+} \nabla\cdot\left(\varepsilon^+\nabla\hat{\psi}^+\right)\mathrm{d}\mathcal{A} + \int_{C_{i,j}\cap\Omega^+}\kappa^2(\mathbf{x})\sinh\left(\hat{\psi}^+\right)\mathrm{d}\mathcal{A} &= \int_{C_{i,j}\cap\Omega^+} f^+\mathrm{d}\mathcal{A},
\end{aligned}$$

where superscript $^+$ represents quantities in Ω^+ and superscript $^-$ represents quantities in Ω^-. Since both equations can be treated in the same way, we only describe the equation in Ω^+ in more detail. By using the divergence theorem on the left-hand side we get the following:

$$-\int_{\partial(C_{i,j}\cap\Omega^+)}\varepsilon^+\nabla\hat{\psi}^+\cdot\mathbf{n}\,\mathrm{d}\ell + \int_{C_{i,j}\cap\Omega^+}\kappa^2(\mathbf{x})\sinh\left(\hat{\psi}^+\right)\mathrm{d}\mathcal{A} = \int_{C_{i,j}\cap\Omega^+} f^+\mathrm{d}\mathcal{A}.$$

The first term in the previous equation can be split into two integrals: the integral over the cell faces inside Ω^+ and the integral over the interface Γ inside the cell. The integration over the cell faces inside Ω^+ is easily approximated numerically using length fractions as demonstrated in [108]. For instance, the integration over the right face is discretized as

$$-\int_{L_R} \varepsilon^+ \nabla \hat{\psi}^+ \cdot \mathbf{n}\, \mathrm{d}\ell = L_{i+1/2,j} \cdot \varepsilon^+ \cdot \frac{\hat{\psi}^+_{i+1.j} - \hat{\psi}^+_{i.j}}{\Delta x} + \mathcal{O}\left(\Delta x^3\right).$$

Accurately integrating over the interface inside the cell is more complicated. This is because, even though the normal to the interface is easily found from the level set function, we do not know the normal derivative of the solution to the interface. To overcome this challenge, one can take the following steps. As shown in figure 3.4, we call point P to be the projection of node (i,j) on the interface Γ, and δ its distance to the interface. Using a first-order Taylor expansion, the jump in the normal derivative, at point P, may be written as:

$$\begin{aligned} \frac{\partial \hat{\psi}^+}{\partial n} - \frac{\partial \hat{\psi}^-}{\partial n} &= \frac{\hat{\psi}^+_P - \hat{\psi}^+_{i,j}}{\delta} - \frac{\hat{\psi}^-_P - \hat{\psi}^-_{i,j}}{\delta} + \mathcal{O}(\Delta x) \\ &= \frac{a_P}{\delta} - \frac{\hat{\psi}^+_{i,j} - \hat{\psi}^-_{i,j}}{\delta} + \mathcal{O}(\Delta x), \end{aligned} \tag{3.7}$$

where a_p denotes the value of the a at node P. Also, the jump in the normal component of electric displacement field, $[\varepsilon \nabla \hat{\psi} \cdot n]$, at node P can be expressed as:

$$\varepsilon^+ \frac{\partial \hat{\psi}^+}{\partial n} - \varepsilon^- \frac{\partial \hat{\psi}^-}{\partial n} = b_P. \tag{3.8}$$

Equations (3.7) and (3.8), are two equations for the two unknowns, $\partial \hat{\psi}^+ / \partial n$ and $\partial \hat{\psi}^- / \partial n$, and yield,

$$\frac{\partial \hat{\psi}^+}{\partial n} = \frac{1}{\delta \left(\varepsilon^- - \varepsilon^+\right)} \left(-\delta b_P + \varepsilon^- \left(a_P - \left(\hat{\psi}^+_{i,j} - \hat{\psi}^-_{i,j}\right)\right)\right) + \mathcal{O}(\Delta x), \tag{3.9}$$

$$\frac{\partial \hat{\psi}^-}{\partial n} = \frac{1}{\delta \left(\varepsilon^- - \varepsilon^+\right)} \left(-\delta b_P + \varepsilon^+ \left(a_P - \left(\hat{\psi}^+_{i,j} - \hat{\psi}^-_{i,j}\right)\right)\right) + \mathcal{O}(\Delta x). \tag{3.10}$$

Once the normal derivatives are found, integration over the interface is approximated as,

$$- \int_{\Gamma \cap C_{i,j}} \varepsilon^+ \frac{\partial \hat{\psi}^+}{\partial n} = |\Gamma| \frac{\varepsilon^+}{\delta \left(\varepsilon^- - \varepsilon^+\right)} \left(\delta b_P + \varepsilon^- \left(\hat{\psi}^+_{i,j} - \hat{\psi}^-_{i,j} - a_P\right)\right) + \mathcal{O}(\Delta x^2), \tag{3.11}$$

and,

$$-\int_{\Gamma\cap C_{i,j}} \varepsilon^- \frac{\partial\hat{\psi}^-}{\partial n} = |\Gamma| \frac{\varepsilon^-}{\delta\left(\varepsilon^- - \varepsilon^+\right)} \left(-\delta b_P - \varepsilon^+ \left(\hat{\psi}^-_{i,j} - \hat{\psi}^+_{i,j} - a_P\right)\right) + \mathcal{O}(\Delta x^2). \tag{3.12}$$

This sharp, first-order discretization leads to a system that has twice as many equations as computational nodes near the interface to account for the jump in the solution and its gradient. It is, however, symmetric positive definite and only involves first-degree neighbors. A major advantage of the scheme is that large differences in ε^+ and ε^- do not adversely affect the accuracy.

3.6 Numerical examples

In this part we provide numerical examples to support the accuracy and convergence of the method. The first example is simply intended to show that the level set method can accurately represent the complicated surface of complex proteins. The next two examples are pure mathematical examples built to demonstrate the accuracy of the method in imposing the jump conditions as described in section 3.5. Next, we consider the Kirkwood's dielectric sphere problem [73], a physically meaningful problem that has an analytic solution. In example 5, the solvation free energy is computed and the results are compared with the APBS software for a select number of proteins [10]. Finally, we conclude this chapter by solving the electrostatic potential on a DNA (1D65) molecule.

3.6.1 Surface accuracy

In this section, we will shortly comment on the accuracy of our method for generating the biomolecular surfaces. To do this, we consider two different studies. First, we consider a "simple" molecule made of three atoms of radii 2 Å, placed on the vortices of an equilateral triangle with the inter-atomic distance of 3 Å. The SES for this molecule is then generated using our new approach and compared with both analytical and triangulated representations obtained via the MSMS software [131] with the probe radius of 1.5 Å. The triangulated surface is generated with the vertex density of 100.

Table 3.6.1 illustrates the accuracy analysis of this test when the grid is refined. We consider two measures to evaluate the accuracy. The forth column depicts the maximum distance of the triangulated surface vertices to the zero level set, i.e our representation of the SES. This is a "local" accuracy measure. The second measure is the comparison between the surface area of the zero level set, denote as A_{LS}, and the exact analytical area computed via MSMS software, denoted as A_{Ex}. This is a more global measure. It is easy to see that the new approach can generate accurate representation of the SES when compared to the MSMS software and the error in the local measure decays linearly with grid spacing. We use the second-order accurate geometric approach of Min and Gibou [100] to compute the areas.

$(res_{\max}, res_{\min})$	# of Points	$\Delta x_{\min}$	$\lvert\phi_{TS}\rvert_{\max}$	Order	$\lvert A_{LS} - A_{Ex}\rvert / A_{Ex}$	Order
(32, 8)	6 393	0.47	1.12×10^{-1}	–	2.00×10^{-2}	–
(64, 16)	46 601	0.23	5.98×10^{-2}	0.91	6.06×10^{-3}	1.80
(128, 32)	353 025	0.12	3.15×10^{-2}	0.92	1.79×10^{-3}	1.76
(256, 64)	2 746 817	0.06	1.58×10^{-2}	1.00	6.24×10^{-4}	1.52
(512, 128)	21 668 481	0.03	7.92×10^{-3}	1.00	2.67×10^{-4}	1.22

Table 3.1: Point-wise accuracy analysis of SES generation for a 3-atom molecule. (Copyright Notice: First published in [103] by Global Science Press.)

As the second test, we choose a set of proteins, compute the total surface area of the SES, and compare our results to those obtained through the MSMS software. For these tests, a probe radius of 1.5 Å has been chosen and all level set functions have been obtained on an adaptive grid with $(res_{\max},\ res_{\min}) = (512,\ 32)$ while leaving parameters of the MSMS software to defaults. Table 3.6.1 illustrates the accuracy of our method by comparing surface area calculation results (in Å^2) with those of the MSMS software (triangulated) and analytical calculations (analytic). The results obtained here indicate that level set method can be used to easily generate accurate molecular surfaces without the need for explicit handling of geometric singularities.

Protein ID	# of Atoms	Analytical	Triangulated	Present work	# of Points	$\Delta x_{\min}$
1AJJ	513	2 112.84	2 013.42	2 131.66	2 075 015	0.101
6RXN	667	2 341.41	2 226.35	2 350.71	2 488 633	0.097
2ERR	1 638	5 189.28	4 917.44	5 074.78	1 373 760	0.190
1AA2	1 755	4 891.97	4 749.54	4 826.41	2 091 216	0.151
2X6A	4 294	13 612.07	12 975.43	13 174.53	1 295 583	0.313
2TEC	4 936	10 568.21	10 072.36	10 187.27	2 253 537	0.210

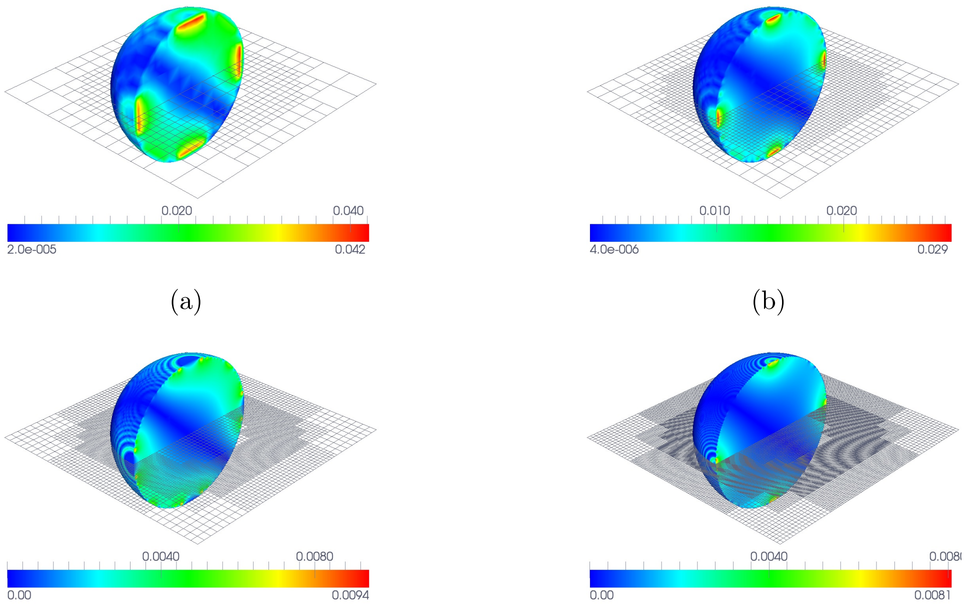
Table 3.2: Comparison of total surface area between the present method and the MSMS software. (Copyright Notice: First published in [103] by Global Science Press.)

3.6.2 Sphere example

In this example, we consider a spherical interface in three spatial dimensions. We take the exact solution to be $\psi^+(x,y,z) = x^3 + y^3 + z^3$ and $\psi^-(x,y,z) = -1 - x^3 - y^3 - z^3$, where $\varepsilon^+ = 2$ and $\varepsilon^- = 80$. The radius of the sphere is 1 and the domain is $[-2,2]^3$. Convergence results given in table 3.6.2, indicate that our method is first-order accurate in L_1, L_2 and L_∞ norms. The resulting numerical error on the different grids used in this analysis is shown in figure 3.5.

$res_{\max}$	Grid points	L_1 Error	Order	L_2 Error	Order	L_∞ Error	Order
32	12 739	5.98×10^{-3}	-	8.71×10^{-3}	-	4.22×10^{-2}	-
64	92 965	2.00×10^{-3}	1.58	3.20×10^{-3}	1.44	2.85×10^{-2}	0.57
128	708 745	7.33×10^{-4}	1.45	1.29×10^{-3}	1.31	9.42×10^{-3}	1.60
256	5 531 665	3.27×10^{-4}	1.16	6.64×10^{-4}	0.96	8.08×10^{-3}	0.22

Table 3.3: Convergence analysis for example 3.6.2. (Copyright Notice: First published in [103] by Global Science Press.)



(a)

(b)

(c)

(d)

Figure 3.5: L_∞ error plotted for the example 3.6.2 on the different octree grids considered in the convergence analysis. Maximum grid resolution in each of figures are (a): $res_{\max} = 64$, (b): $res_{\max} = 128$, (c): $res_{\max} = 256$, (d): $res_{\max} = 512$. (Copyright Notice: First published in [103] by Global Science Press.)

3.6.3 A test biomolecule

In this example, we consider the same parameters as in the previous example, except that now the interface is the 2ERR molecule. Convergence results are given in table 3.6.3. Our approach still appears to be first-order in L_∞ norm, but second-order accurate in L_1 and L_2 norms. The numerical error is represented in figure 3.6.

$res_{\max}$	Grid points	L_1 Error	Order	L_2 Error	Order	L_∞ Error	Order
64	19 602	6.00×10^{-3}	-	7.43×10^{-3}	-	2.11×10^{-2}	-
128	146 087	1.41×10^{-3}	2.09	1.73×10^{-3}	2.10	7.2×10^{-3}	1.55
256	1 124 793	3.37×10^{-4}	2.06	4.15×10^{-4}	2.06	5.69×10^{-3}	0.34
512	8 871 297	8.76×10^{-5}	1.94	1.11×10^{-4}	1.90	3.13×10^{-3}	0.86

Table 3.4: Convergence analysis for example 3.6.3. (Copyright Notice: First published in [103] by Global Science Press.)

3.6.4 Kirkwood's solution

Here we consider a more physically relevant example; the Kirkwood's dielectric sphere as presented in [73]. Consider an ionic solution with dielectric constant ε^-. In the electrolyte, a sphere of radius b is placed with dielectric constant ε^+, which we take to be unity. Inside the sphere, we consider M discrete point charges, $q_1 \cdots q_M$. Using the spherical coordinate system, with the origin at the center of the sphere, the solution to the electrostatic potential, for $r < b$, is given by

$$\begin{aligned} V_1 &= \sum_{k=1}^{M} \frac{q_k}{4\pi\varepsilon^+ |\mathbf{r} - \mathbf{r}_k|} + \psi, \\ \psi &= \sum_{n=0}^{\infty} \sum_{m=-n}^{n} B_{mn} r^n P_n^m (\cos\theta)\, e^{im\phi}, \end{aligned}$$

where $|\mathbf{r} - \mathbf{r}_k|$ is the distance from the charge q_k and ψ is the contribution to the potential from the charge distribution in the surrounding electrolyte. The functions P_n^m are the associated Legendre polynomials. The potential in the shell, $b < r < a$, is given by

$$V_2 = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \left(\frac{C_{mn}}{r^{n+1}} + G_{mn} r^n \right) P_n^m (\cos\theta)\, e^{im\phi},$$

and the potential in the electrolyte solution, $r > a$, is given by

$$\begin{aligned} V_3 &= \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \left(\frac{A_{mn}}{r^{n+1}} \right) K_n (\kappa r) \exp(-\kappa r) P_n^m (\cos\theta)\, e^{im\phi}, \\ K_n (x) &= \sum_{s=0}^{n} \frac{2^s n! \,(2n-s)!}{s! \,(2n)! \,(n-s)!} x^s, \end{aligned}$$

where κ is the usual Debye parameter. The coefficients A_{mn}, B_{mn}, C_{mn}, and G_{mn} are found, for each set of m and n, by imposing the interface jump conditions.

The interested reader is referred to [73] for more details on the derivation of these equations.

The Kirkwood sphere chosen here has a radius of 30 Å, a shell thickness of 3 Å and one negative charge in the middle of the sphere. Moreover, a 1 mM electrolyte was chosen with $\varepsilon^+ = 2$ and $\varepsilon^- = 80$. The convergence results provided in table 3.6.4, clearly demonstrate that our method is at least first-order. The same analysis is also performed close to the boundary. The results, presented in table 3.6.4, suggest that indeed the maximum error appears near the boundary, where the jump conditions are imposed, and thus the idea of grid refinement near the boundary is justified.

$res_{\max}$	Grid points	L_1 Error	Order	L_2 Error	Order	L_∞ Error	Order
64	18 747	5.61×10^{-3}	-	1.13×10^{-2}	-	5.02×10^{-2}	-
128	138 005	1.09×10^{-3}	2.36	2.72×10^{-3}	2.05	1.09×10^{-2}	2.20
256	1 057 257	3.11×10^{-4}	1.81	8.83×10^{-4}	1.62	3.47×10^{-3}	1.65
512	8 273 105	1.01×10^{-4}	1.62	3.32×10^{-4}	1.41	1.23×10^{-3}	1.49

Table 3.5: Convergence analysis for the Kirkwood's solution 3.6.4 in the whole domain. (Copyright Notice: First published in [103] by Global Science Press.)

$res_{\max}$	Grid points	L_1 Error	Order	L_2 Error	Order	L_∞ Error	Order
64	18 747	4.26×10^{-2}	-	4.32×10^{-2}	-	5.02×10^{-2}	-
128	138 005	8.31×10^{-3}	2.36	8.43×10^{-3}	2.34	1.09×10^{-2}	2.20
256	1 057 257	2.37×10^{-3}	1.81	2.44×10^{-3}	1.78	3.47×10^{-3}	1.65
512	8 273 105	7.68×10^{-4}	1.63	8.06×10^{-4}	1.60	1.23×10^{-3}	1.49

Table 3.6: Convergence analysis for the Kirkwood's solution 3.6.4 near the boundary. (Copyright Notice: First published in [103] by Global Science Press.)

3.6.5 Solvation free energy

In this section, we provide numerical examples that illustrate the accuracy of our method in computing the electrostatic solvation free energy of certain proteins. The total electrostatic free energy of a biomolecule is given by [135]:

$$\Delta G^{\mathrm{T}} = \int_{\Omega} \left(\rho^{\mathrm{f}} \psi - 2 k_{\mathrm{B}} T c_{\infty} \left(\cosh \left(\frac{\psi e}{k_{\mathrm{B}} T} \right) - 1 \right) - \frac{\varepsilon^{-}}{2} \left| \nabla \psi \right|^{2} \right) \, \mathrm{d}\mathcal{V},$$

where ρ^{f} is the charge density of the singular charges. In this equation, the first, second and last term correspond to the energy due to singular charge interactions, osmotic pressure and the energy stored in the electric field. Since the existence of a jump in the dielectric coefficient indicates a discontinuity in the electric field, it is desired to replace the last term by an equivalent term that only depends on the potential itself and not its gradient. This is easily done by incorporating the Poisson-Boltzmann equation [97] which results in the following equation:

$$\begin{aligned} \Delta G^{\mathrm{T}} &= \int_{\Omega} \left(\frac{1}{2} \rho^{\mathrm{f}} \psi - 2 k_{\mathrm{B}} T c_{\infty} \left(\cosh \left(\frac{\psi e}{k_{\mathrm{B}} T} \right) - 1 \right) + \psi e c_{\infty} \sinh \left(\frac{\psi e}{k_{\mathrm{B}} T} \right) \right) \, \mathrm{d}\mathcal{V} \\ &- \frac{1}{2} \int_{\partial\Omega} \varepsilon^{-} \frac{\partial \psi}{\partial n} \psi \, \mathrm{d}\mathcal{A}. \end{aligned}$$

To obtain the solvation free energy, it is required that we subtract the energy associated with a reference state. This reference state is taken to be the energy stored in the electric field that is only due to the singular charges in a medium with the same dielectric constant as that of the molecule. Using Green functions, this energy is given by

$$\Delta G^{\mathrm{ref}} = \frac{1}{2}\sum_{i=1}^{N_m}\sum_{\substack{j=1\\ j\neq i}}^{N_m} \frac{q_i q_j}{4\pi\varepsilon^{+}\left|\mathbf{r}_i - \mathbf{r}_j\right|},$$

where q_i is the charge of the i-th atom, ε^{+} is the dielectric constant of the molecule, N_m is the total number of atoms in the molecule and $\mathbf{r}_i$ is the position vector of the center of the i-th atom. Using the decomposition for the potential, as described in section 3.4.1, and the fact that $\psi = 0$ on $\partial\Omega$, the final form of solvation free energy may be written as:

$$\begin{aligned}\Delta G^{\mathrm{sol}} &= \Delta G^{\mathrm{T}} - \Delta G^{\mathrm{ref}} \\ &= \frac{1}{2}\sum_{i=1}^{N_m} q_i\left(\psi^0 + \hat{\psi}\right) \\ &- 2k_{\mathrm{B}}Tc_{\infty}\int_{\Omega^-}\left(\cosh\left(\frac{\hat{\psi}e}{k_{\mathrm{B}}T}\right) - 1\right)\mathrm{d}\mathcal{V} \\ &+ ec_{\infty}\int_{\Omega^-}\hat{\psi}\sinh\left(\frac{\hat{\psi}e}{k_{\mathrm{B}}T}\right)\mathrm{d}\mathcal{V}. \qquad (3.13)\end{aligned}$$

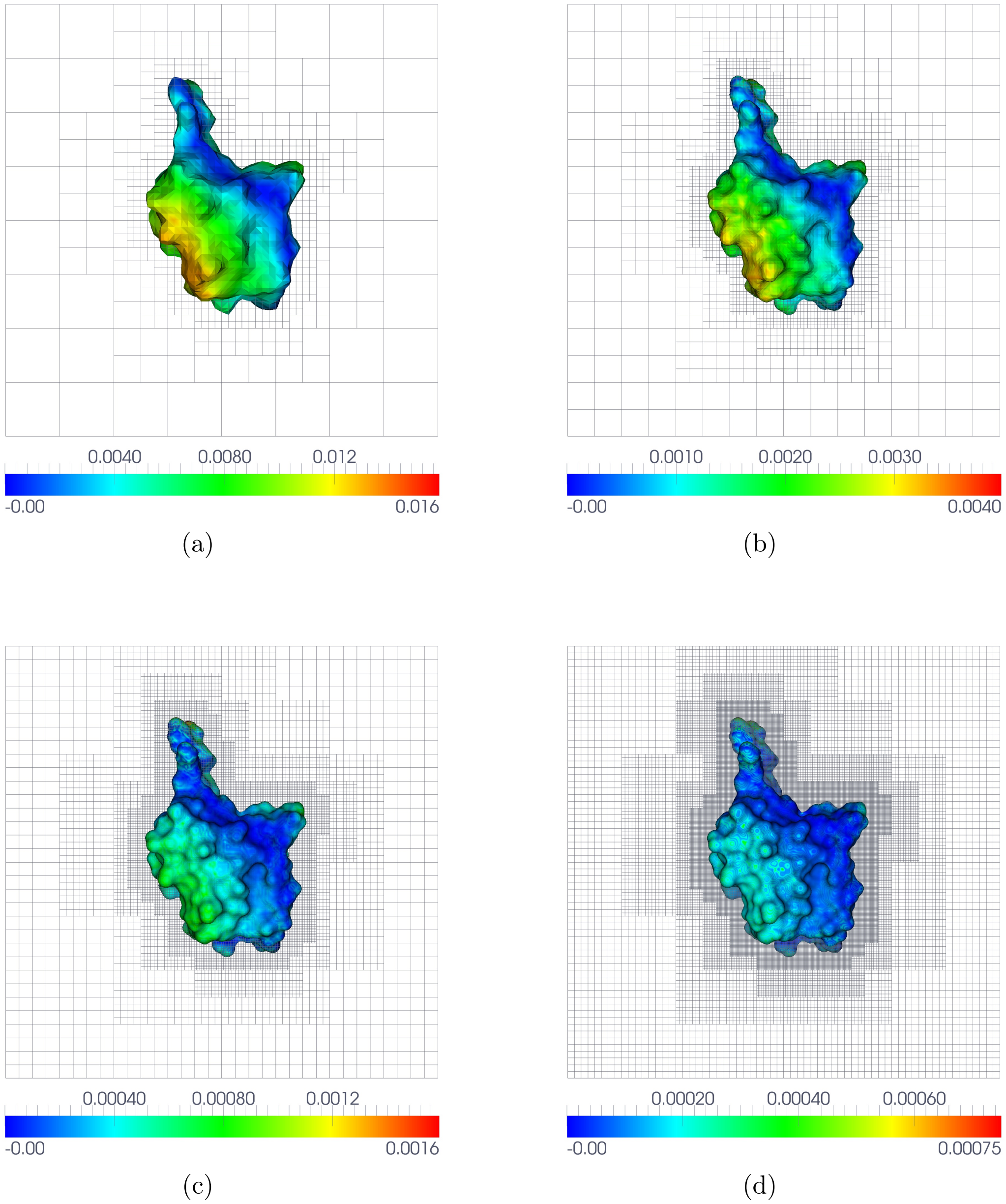
To compare our results, we have used the APBS software to solve the same problem. In all of our calculations we have used an adaptive octree grid with $(res_{\max}, res_{\min}) = (512,\ 32)$. As for physical parameters, we set the bulk concentration to 10 mM and the dielectric coefficients to $\varepsilon^+ = 2$ and $\varepsilon^- = 78.3$ for the molecule and electrolyte, respectively. Table 3.6.5 illustrates the computed solvation free energy, using equation (3.13), in kJ/mol and compares the results with energies computed using APBS software. It is easily seen that the presented method is capable of producing accurate results even for complicated proteins.

Protein ID	# of Atoms	ΔG^{sol}	$\Delta x_{\min}$	ΔG^{sol} (APBS)	Δx (APBS)	Rel. difference
1AJJ	513	-2.234×10^3	0.14	-2.228×10^3	0.52	1.23×10^{-1}
6RXN	667	-2.274×10^3	0.14	-2.313×10^3	0.51	1.71×10^{-2}
2ERR	1 638	-3.859×10^3	0.27	-4.016×10^3	0.56	4.00×10^{-2}
1AA2	1 755	-3.233×10^3	0.19	-3.327×10^3	0.52	2.87×10^{-2}
2X6A	4 294	-5.388×10^3	0.44	-5.699×10^3	0.71	5.61×10^{-2}
2TEC	4 936	-4.227×10^3	0.30	-4.486×10^3	0.71	5.96×10^{-2}

Table 3.7: Solvation free energy. (Copyright Notice: First published in [103] by Global Science Press.)

3.6.6 Electrostatic potential on a DNA strand

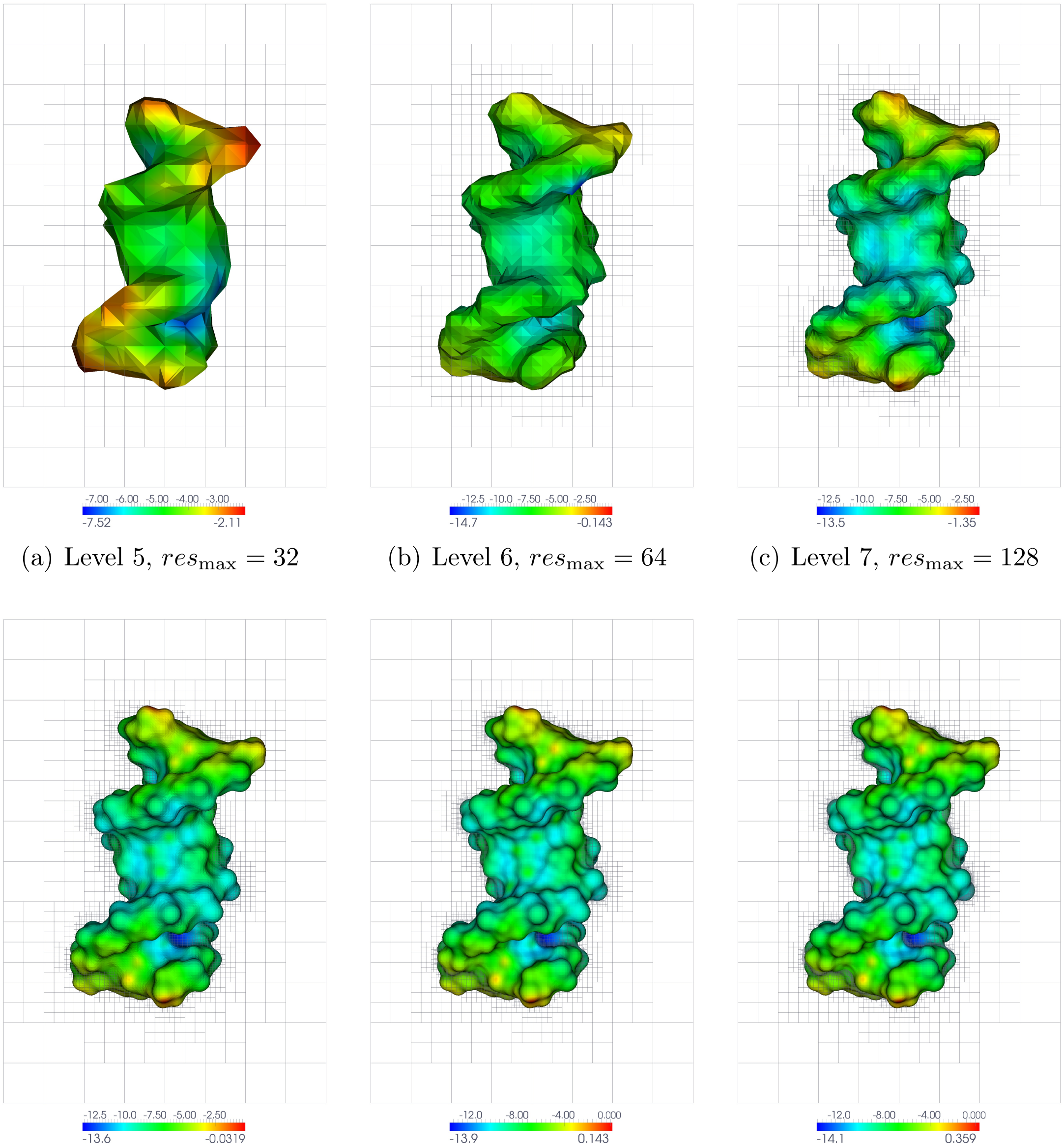
In this last section we illustrate the results of a Poisson-Boltzmann computation for a DNA strand (1D65). Figure 3.7 illustrates the electrostatic potential, in units of thermal voltage, on the surface of the molecule. It is interesting to note how the shape of the protein and the electrostatic potential are affected as the grid is refined. As shown in figure 3.7, coarser grids can only capture the overall shape of the protein whereas more details are only obtained on octree grids with higher resolutions. This, indeed, is a good example that illustrates certain levels of accuracy are only attainable with very high levels of refinement. One should note that this level of resolution ($res_{\max} = 1\ 024$ for level 10) is only feasible on an adaptive grid. Where a uniform grid of the same maximum resolution would require about one billion grid points, this calculation was made on an adaptive octree grid with only about four million grid points ($\sim 0.4\%$).



(a) (b)

(c) (d)

Figure 3.6: L_∞ error plotted for the example 3.6.3 on the different octree grids considered in the convergence analysis. Grid resolutions of each subfigure are (a): $res_{\max} = 64$, (b): $res_{\max} = 128$, (c): $res_{\max} = 256$, (d): $res_{\max} = 512$. (Copyright Notice: First published in [103] by Global Science Press.)



(a) Level 5, $res_{\max} = 32$

(b) Level 6, $res_{\max} = 64$

(c) Level 7, $res_{\max} = 128$

(d) Level 8, $res_{\max} = 256$

(e) Level 9, $res_{\max} = 512$

(f) Level 10, $res_{\max} = 1\ 024$

Figure 3.7: The electrostatic potential on a DNA strand for different levels of refinement. (Copyright Notice: First published in [103] by Global Science Press.)

Chapter 4

Discretization of Poisson-Nernst-Planck Equations

4.1 Introduction

At the continuum level, aqueous electrolytes (e.g. salty water) can be described as conducting liquids. In such a description, the solvent phase is treated as a standard continuum and discrete ions are replaced by continuous concentration fields. Mutual ion-ion interactions, whether steric or electrostatic, are replaced by interactions between ions and a mean-field potential. The simplest of such models is described by the so-called Poisson-Nernst-Planck (PNP) equations [129, 93]. The two most important assumptions used in deriving the PNP equations are:

1. ions are effectively point size and thus all steric interactions, due to finite ion size, are absent,

2. ions only interact via a mean-field electrostatic potential field.

Although these assumptions are violated in certain cases [72, 14], PNP equations are still very useful in studying many electrochemical and biological phenomena. Examples include, but are not limited to, colloid and interface sciences [129, 93], electro-osmosis in micro- and nano-fluidic systems [139, 138], electrophoretic motion of charged particles [112, 93], computation of solvation energies of biomolecules [89], electro-kinetics of electro-chemical cells [16] and in the study of supercapacitors [45].

Although our algorithm is quite general and can be applied, at least partially, to many of the application mentioned above, our focus in this work is on the study of supercapacitor (c.f. figure 4.1). In designing supercapacitors, it is important to know the effect of the pore micro-structure; in particular its effect on the charging capacity and the charging time of the device. First theoretical studies of supercapacitors date back to the Transmission-Line (TL) model [51]. The TL model is effectively a one-dimensional, equivalent RC circuit model for individual pores in a porous electrode where each pore is represented as a perfect cylinder. Since its introduction, this linear model has been extensively used in studying [122, 75, 69] and optimizing supercapacitors [54, 110]. Despite its widespread use, the TL model faces serious shortcomings. First, the TL theory is only valid at very small voltages (< 25 mV). In practice, however, applied potentials are typically around 1-2.5 volts, depending on the electrolyte used [75]. Second, the

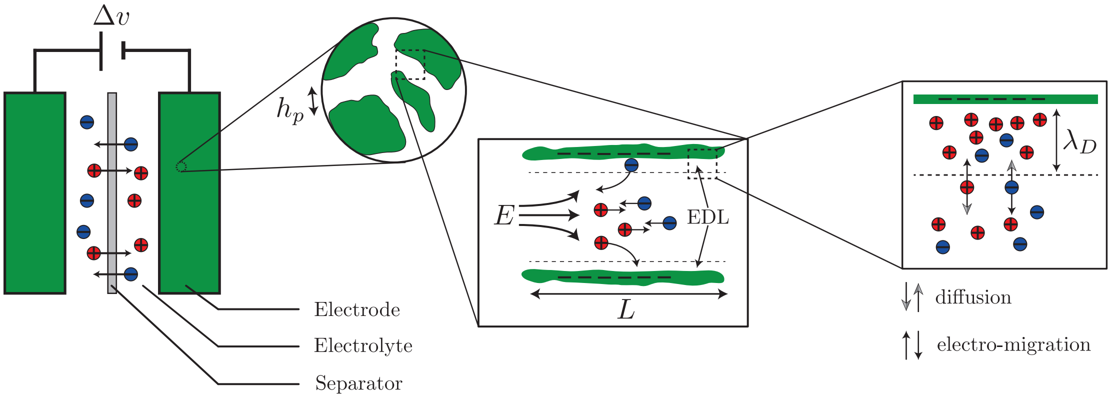


Figure 4.1: Schematic of a supercapacitor. A supercapacitor is made of two porous electrodes (typically carbon) that are in contact with an electrolyte medium and separated by a separator permeable to ions. Once a potential difference is applied to the electrodes, an electric field is set up that drives ions inside the electrolyte and through pores. This current charges the electrodes by bringing "counter-ions" near to the surface and driving "co-ions" away from the surface. This effectively "screens" the charge on the electrode by forming the Electric Double Layer (EDL) near the electrode surface. At any given section along the pore, the charging process continues until the diffusion and electro-migration of ions locally reach a balance (see section 4.2). At steady state, the work by the external power source is electrostatically stored inside the EDL similar to normal capacitors.

TL model assumes that the local EDL thickness is much smaller than the pore radius. This condition, although satisfied for wider macro-pores, does not hold at the nano-pores where pore sizes are on the order of, or even smaller than, the Debye layer [136]. Finally, the actual pore micro-structure are explicitly ignored in TL model.

It is possible to address some of the shortcomings of the TL model, e.g. the low voltage applicability and, to some extend, the effects of micro-structure geometry, by utilizing more sophisticated volume-averaging techniques as initially proposed by [107], and later further developed by Bazant and Biesheuvel [20]. Specifically the model proposed in [20], which we shall refer to the BB model from now on, has been recently used to study various aspects of supercapacitors and similar technologies [126, 156, 143]. Although the BB model is generally a more realistic model than the TL model, it is nonetheless a volume-averaged model and thus has certain shortcomings. For instance, the averaging implicitly assumes a fairly isotropic porous structure; any anisotropy could lead to uneven transport in different directions. Moreover, like the TL model, the BB model is essentially built on the assumption that the EDL is thin compared to a "reference pore dimension". Finally, effects of surface conduction phenomena [93] are completely ignored, *assuming* they are negligible [20].

Despite their popularity, and known shortcomings, very few attempts have been made to directly validate the aforementioned models against a full scale direct numerical simulation (DNS) of the PNP equations at the pore scale. Even the few existing studies are conducted at low voltages; either completely [130] or implicitly for the most part [84]. This is likely due to the numerical challenges associated with the different length scales inherent in the PNP equations at high voltages. In particular, concentration fields often decay exponentially with the local electric potential, which itself decays exponentially with the distance from wall, with a length scale proportional to the EDL thickness [129]. Any successful algorithm must therefore be sophisticated enough to handle such steep gradients efficiently. The PNP equations also involve nonlinear terms describing the flux of ions generated by electric fields. Finally, a useful algorithm, needs to be able to handle the arbitrary and complicated geometry of pore micro-structures (see figure 4.1).

In terms of algorithm development for solving the PNP equations, the biophysics community has been very active. For instance, one-dimensional finite difference methods were utilized by many authors for simulating ion-selective membranes and cellular ion-channels [41, 27, 82, 55, 34]. Two and three-dimension algorithms have also been reported in the literature. In [77, 33] authors developed a finite difference, SOR-like algorithm for the steady-state PNP equations. Transient finite difference [159] and finite element [89] methods have also been developed with biophysical applications in mind; though they are general enough that it should be relatively straightforward to apply them to other problems as well. Other communities, such as electro-chemistry and fluid dynamics, specially since the advent of micro- and nano-fluidic devices, have also been active in this field. For instance, simple one-dimensional finite difference methods have been proposed for studying different aspects of electrochemical cells [104, 95, 16]. Two- and three-dimensional methods have also been proposed, either for, more or less, general-purpose calculations [78, 12] or specific to a particular problem, such as copper electro-deposition [29, 30], concentration polarization shocks [155], or electro-kinetic instability [85, 52]. For a broader overview of proposed algorithms, we refer the interested reader to a recent review article [149].

Given the existing rich literature, the most distinct feature of this work is the use of adaptive quadtree grids to efficiently handle large gradients in the EDL. This is based on successful application of quadtree grids for solving the Poisson-Boltzmann equations as presented in previous chapters [102**?**]. However, as discussed in section 4.3, this extension requires a novel hybrid approach to ensure that the resulting algorithm is conservative and spurious numerical fluxes do not adversely affect the accuracy of computed solution.

This chapter is organized as follows: section 4.2 describes the PNP equations and the relevant boundary conditions. Standard non-dimensionalization demonstrates the dependence on two nondimensional variables, namely the electrode potential and the nondimensional EDL thickness. In section 4.3.1, we describe how the physical boundary (e.g. of an electrode) is handled via the level-set framework and discuss several strategies that can be used to automatically generate quadtree adaptive grids that can efficiently resolve large gradients and fluxes in the EDL. In section 4.3.2 and 4.3.3, we describe the temporal and spatial discretizations in two spatial dimensions. In section 4.4, we present several numerical experiments that demonstrate the second-order accuracy in space and first-order accuracy in time. Moreover we explicitly demonstrate the conservativeness properties of our method.

4.2 Poisson-Nernst-Planck theory

The Poisson-Nernst-Planck (PNP) theory is a simple, continuum-based approach towards modeling the dynamics of ions and evolution of electric field inside electrolyte solutions. The PNP theory is derived from the fundamental transport equations: consider a binary electrolyte that, when dissolved in water, dissociates into cations (positive ions) and anions (negative ions) that fill the entire solution. As such, it is possible to describe the electrolyte solution via two scalar fields, c_+ and c_-, that represent the concentration of cations and anions, respectively. Furthermore, it is assumed that all electrostatic interactions between ions may be reduced to Coulombic interactions between ions and a mean-field electric potential, ψ. Conservation of ions then requires that [129, 93]:

$$\partial_t c_i = -\nabla \cdot \mathbf{J}_i,$$

where $\mathbf{J}_i$ is the total flux of ion species, $i = \pm$, and, in the absence of fluid motion, is given by

$$\mathbf{J}_i = -D_i \nabla c_i - \mu_i z_i e \nabla \psi \; c_i.$$

In this equation, the first term represents the familiar diffusive flux and D_i is the diffusion coefficient. The second term is the electro-migrative flux that accounts for the motion of ions due to electrostatic interactions between the electric field, $\mathbf{E} = -\nabla \psi$, and the electric charge on the ion, $z_i e$, where z_i is the ion valency and e is the fundamental charge of a proton. Finally μ_i is the mobility coefficient that relates the drift velocity of ions to the electric force exerted on them. Mobility coefficient is related to diffusion coefficient via Einstein's relation $\mu_i = D_i / k_{\mathrm{B}} T$ where k_{B} and T are the Boltzmann constant and absolute temperature [49], respectively. The mean-field potential, ψ, is related to the ion concentration via the Poisson's equation:

$$\nabla \cdot \varepsilon \nabla \psi = -\sum_i z_i c_i e,$$

where ε is the permitivity coefficient of the electrolyte.

Assuming a binary, symmetric z:z electrolyte with constant diffusivity and permittivity coefficients, it is possible to nondimensionalize the PNP equations by introducing the following nondimensional variables:

$$t^* = \frac{tD}{L^2}, \qquad \mathbf{x}^* = \frac{\mathbf{x}}{L}, \qquad c_i^* = \frac{c_i}{2c_\infty}, \qquad \psi^* = \frac{\psi \, ez}{k_{\mathrm{B}}T},$$

where L and c_∞ are referrence length and concentration scales, respectively. Rewritting the PNP equations in terms of non-dimensionalized variables, after dropping all star superscripts, yields:

$$\partial_t c_+ = \nabla^2 c_+ + \nabla \cdot c_+ \nabla\psi \tag{4.1}$$

$$\partial_t c_- = \nabla^2 c_- - \nabla \cdot c_- \nabla\psi \tag{4.2}$$

$$\nabla^2\psi = -\kappa^2 \left(c_+ - c_-\right) \tag{4.3}$$

where $\kappa = L/\lambda_D$ is the ratio of the reference length scale to the Debye layer thickness. As detailed in previous chapters, the Debye layer thickness is defined as [129]:

$$\lambda_D = \sqrt{\frac{\varepsilon k_{\mathrm{B}} T}{2c_\infty e^2 z^2}},$$

and represents the natural length scale over which electrostatic charges on the wall are screened by the ions in the electrolyte and is typically of the order of few nanometers (see figure 4.1).

Various boundary conditions may be used with the PNP equations. Here we consider the simplest form of boundary conditions for the concentration and for the electric potential field. For the concentration, one usually ignores any chemical reactions at the solid boundaries and assumes a no flux condition, i.e.

$$\begin{aligned} \hat{\mathbf{n}} \cdot \nabla c_{+} + \hat{\mathbf{n}} \cdot \nabla \psi \; c_{+} &= 0 \qquad (4.4) \\ \hat{\mathbf{n}} \cdot \nabla c_{-} - \hat{\mathbf{n}} \cdot \nabla \psi \; c_{-} &= 0 \qquad (4.5) \end{aligned}$$

However, it should be noted that it is possible to account for reactions through, for example, using the Butler-Volmer reaction kinetics [13]. The boundary conditions for the electrostaic potential is however not unique and greatly depends on the problem under investigation. Usually, one ends up either specifying the potential or the charge density on the boundary, which corresponds to Dirichlet or Neumann boundary conditions, respectively. In general, though, it is possible to use more complex boundary conditions that would, for example, account for effects of dielectric coating or a stern layer [16]. Here we are only concerned with the simple case of a Dirichlet boundary condition for the electric potential, i.e. we impose

$$\psi = v\left(\mathbf{x}\right), \qquad (4.6)$$

on the electrode's boundary, where $v(\mathbf{x})$ is a known function.

4.3 Numerical approach

4.3.1 Grid generation and domain description

Just as in the previous chapters, we choose an implicit surface representation (level-set) to define the electrode's boundary to ease the treatment of boundary conditions on arbitrary surfaces. This also greatly simplifies the task of generating adaptive grids as discussed earlier.

The PNP equations have two length scales that can be identified very easily, namely the EDL scale and the bulk scale. The EDL roughly extends $\mathcal{O}(\kappa^{-1})$ away from the surface and is characterized by very large gradients in ion-concentration and electric potential, whereas these gradients are very small in the bulk. A classical example showing the separation of the two scales is the potential and ion distribution above an infinite planar electrode. At steady state, these are given by the Gouy-Chapman solution [129]:

$$\psi = 4\tanh^{-1}\Big(\tanh\left(\frac{v}{4}\right)\exp\left(-\kappa x\right)\Big), \tag{4.7}$$

$$c_{\pm} = \frac{1}{2}\exp\left(\mp\psi\right), \tag{4.8}$$

where v is the potential on the surface and x is the distance away from it. Note that, although κ^{-1} is the relevant length scale here, it is easy to see that even for moderately large surface potentials, say $v = 5$, the gradient in the counter-ion concentration is even larger than that of the potential due to the exponential dependence of concentration on the potential. It is clear that to efficiently handle this multiple length scale problem it is desirable to resort to adaptive grids.

Many different refinement criteria may be defined. In chapter 2 we presented a grid refinement method according to the distance from the boundary which is easy to implement thanks to the level-set representation of surface. This refinement strategy is justified for PNP equations for the exact same reason it was used in PB calculations, i.e. since the EDL forms close to the interface. Moreover, the EDL thickness, κ^{-1}, can be used as a metric to tune the maximum and minimum grid resolutions close to the boundary. We note, however, that this kind of refinement may not be very efficient or enough to capture all the details inside the EDL. A more conservative approach could be refinement based on the gradients in the electric potential and the concentration fields, either using equations (4.7) and (4.8) as rough estimates or via solving the Poisson-Boltzmann equation for the given geometry. Note that these approaches are simply based on physical arguments about the EDL which may not necessarily result in smaller error. This could happen, for example, if the T-junction nodes are present where large gradients exist inside the EDL. In such cases, it may be beneficial to extend the layer of finest grid cells far enough to encompass large gradients.

Refinement could also be done based on trying to minimize or bound the error in the domain. This could be as simple as using the Gouy-Chapman solutions, (4.7) and (4.8), or by utilizing Richardson's error estimators for the Poisson-Boltzmann equation. Both of these approaches are based on bounding the error at equilibrium. For applications considered in this dissertation, this is a conservative approach since largest gradients only develop toward the end of simulation and at steady state when the EDL is fully charged. However, in other applications with more transient or non-equilibrium nature, such error estimation may not necessarily be accurate. It is possible to apply Richardson error estimators to the transient PNP equations as well. This would truly adapt the grid at each time step but comes at the prohibitive cost of error estimation at every single step. Thus it is not clear if this approach would be any more efficient than the static error estimation.

As can be seen, many different approaches may be taken towards defining the actual refinement criteria. Here we do not discuss the full analysis of these different refinement criteria and simply resort to the simplest criteria described by equation (2.3) in chapter 2.

4.3.2 Time integration

Implicit-Explicit (IMEX) methods [5, 4] are quite popular and provide an effective treatment of time discretizations to problems that have both linear and nonlinear parts. The basic idea behind IMEX methods is to treat linear parts of the problem implicitly, hereby avoiding a stringent step restriction, while treating nonlinear parts explicitly. This is specially useful when the linear part comes from the discretization of diffusion operator and is stiff. In the context of the PNP calculations this is even more important since the existence of the EDL imposes a very small grid resolution, $h_{\min} \ll \kappa^{-1}$, near the boundary. Moreover, treating nonlinear parts explicitly eliminates the need for linearization and thus avoid the costs associated with computing Jacobians. A general IMEX discretization for the concentration fields in PNP equations is given by:

$$\begin{aligned} \sum_{i=0}^{M} \alpha_i c_{\pm}^{n+1-i} &= -\nabla \cdot \tilde{\mathbf{J}}_{\pm}^{n+1} \\ &= \nabla^2 c_{\pm}^{n+1} - \nabla \cdot \tilde{\mathbf{J}}_{e\pm}^{n+1}. \end{aligned}$$

Here, the α_i's are the discretization coefficients and M is the degree of discretization in time. Moreover, $\tilde{\mathbf{J}}_{\mathbf{e}\pm}^{n+1}$ is a consistent approximation to the nonlinear electromigrative flux at time-step t^{n+1}, i.e. it is required that $\tilde{\mathbf{J}}_{e\pm}^{n+1} = \mathbf{J}_{e\pm}^{n+1} + \mathcal{O}\left(\Delta t^M\right)$.

Extrapolation is utilized to satisfy this condition, i.e. we have

$$\tilde{\mathbf{J}}_{e\pm}^{n+1} = \sum_{i=0}^{M-1} \beta_i \mathbf{J}_{e\pm}^{n-i},$$

where the β_is are extrapolation coefficients.

Here we will consider the simplest case of time discretizations, i.e. with $M = 1$. This simply corresponds to a first-order semi-implicit Euler method with $\alpha_0 = 1/\Delta t$, $\alpha_1 = -1/\Delta t$, and $\beta_0 = 1$. Time discretization for concentration fields simply becomes:

$$\begin{aligned} \frac{c_+^{n+1} - c_+^n}{\Delta t} &= \nabla^2 c_+^{n+1} + \nabla \cdot c_+^n \nabla \psi^n, && (4.9) \\ \frac{c_-^{n+1} - c_-^n}{\Delta t} &= \nabla^2 c_-^{n+1} - \nabla \cdot c_-^n \nabla \psi^n. && (4.10) \end{aligned}$$

Finally, since the Poisson equation is not time dependent it is simply solved after the concentration fields are computed at the new time step:

$$-\nabla^2 \psi^{n+1} = \kappa^2 \left(c_+^{n+1} - c_-^{n+1} \right). \tag{4.11}$$

4.3.3 Spatial discretization

Physically, the Poisson equation (4.3) is very different from Nernst-Planck equations (4.2) and (4.3). Specifically, the Nernst-Planck equations describe conservation laws for ion concentrations whereas the Poisson equation, in the context of PNP equations, does not have this property. As a result it is desirable, and at large voltages necessary, that the discrete form of the Nernst-Planck equations also conserves the mass both locally and globally.

The nodal discretizations on quadtree adaptive grids of [101, 35, 60] do not satisfy conservation properties at T-junctions but produce second-order accurate solutions and second-order accurate approximations of gradients. As a result nodal approximations are appropriate, and desirable, for the spatial discretization of the Poisson equation. We will therefore utilize the same nodal approach for the Poisson equation. Since the discretizations along with the treatment of boundary condition is very similar to the Poisson-Boltzmann equation, we do not repeat the discretization technique and refer the reader to section 2.3.2 for the details.

As mentioned earlier, large gradients exist inside the EDL, especially at large voltages. Since the Nernst-Planck equations describe conservation properties, it is desired to utilize conservative discretizations when possible. This is specially important since T-junctions generally exist inside EDL where gradients are large and non-conservative properties of nodal discretizations are amplified. Various conservative methods on adaptive Cartesian grids have been proposed, most notably for discretization of Hyperbolic Conservation Laws [19, 18] and Navier-Stokes equations [3, 88, 87]. Here, the method described in [87] is chosen for the discretization of the diffusion part whereas the electro-migrative part is discretized using a novel hybrid approach, detailed in section 4.3.3.2.

4.3.3.1 Diffusion

Since the discretization of equations (4.9) and (4.10) are similar, we only consider the discretization of the equation for the cations and drop the + sign for brevity. Let us consider the concentration variable, c, that is stored at the center of cell $\mathcal{C}_c$ as illustrated in figure 4.2. Integration of the left-hand side of (4.9) is trivial and yields:

$$\int_{\mathcal{C}_c} \frac{c^{n+1} - c^n}{\Delta t} \,\mathrm{d}\mathcal{A} \approx \frac{c^{n+1} - c^n}{\Delta t} \mathcal{A}_c, \tag{4.12}$$

where $\mathcal{A}_c$ is the area of cell $\mathcal{C}_c$ which is simply $\mathcal{A}_c = \Delta x_c \Delta y_c$ if the cell is entirely in Ω^-. If the cell $\mathcal{C}_c$ is crossed by the boundary Γ, we utilize the geometric integration technique introduced in [100] to evaluate the cell area. The diffusion part is discretized using the divergence theorem:

$$\int_{\mathcal{C}_c} \nabla^2 c \,\mathrm{d}\mathcal{A} = \int_{\partial\mathcal{C}_c} \hat{\mathbf{n}} \cdot \nabla c \,\mathrm{d}\ell.$$

Consider the discretization of the flux term at the right boundary, x^+, of a general cell configuration as depicted in figure 4.2(a). Following [87], we write:

$$\int_{x^+} \hat{\mathbf{n}} \cdot \nabla c \, \mathrm{d}\ell \approx \overline{F^d} \, \Delta y_c, \tag{4.13}$$

where $\overline{F^d}$ is the average diffusive flux density across all the cells at the right boundary and is written as:

$$\begin{aligned} \overline{F^d} &= \frac{1}{\Delta y_c} \sum_{j=1}^{N_{x^+}} F_j^d \, \Delta y_j, & (4.14) \\ &= \frac{1}{\Delta y_c} \sum_{j=1}^{N_{x^+}} \frac{c_j - c_c}{\overline{\Delta x}} \, \Delta y_j. & (4.15) \end{aligned}$$

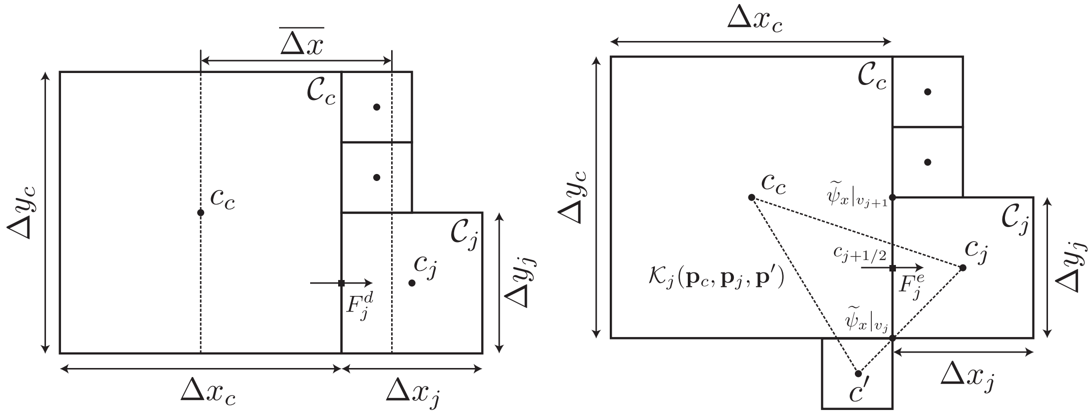
Here, $\overline{\Delta x}$ is the average distance between cell centers and is written as:

$$\overline{\Delta x} = \frac{1}{\Delta y_c} \sum_{j=1}^{N_{x^+}} \frac{\Delta x_c + \Delta x_j}{2} \, \Delta y_j. \tag{4.16}$$

Notice that the definition of $\overline{F^d}$ and $\overline{\Delta x}$ are the same for all the cells that share the right boundary, i.e. the same calculated values are used for both $\mathcal{C}_c$ and all the $\mathcal{C}_j$'s alike. This leads to a conservative symmetric discretization of the diffusion operator by combining equation (4.13) through (4.16):

$$\int_{x^+} \hat{\mathbf{n}} \cdot \nabla c \, \mathrm{d}\ell \approx \frac{2}{\sum_{i=1}^{N_{x^+}} (\Delta x_c + \Delta x_i) \Delta y_i} \sum_{j=1}^{N_{x^+}} \Delta y_c \Delta y_j (c_j - c_c). \tag{4.17}$$

Discretizations along the other sides are similar. Finally, we note that the discretizations in three spatial dimensions are similar with the cell surface area replacing the edge length and the cell total volume replacing the cell total area.



(a) (b)

Figure 4.2: General cell configuration for the spatial discretization of (a) diffusion (see section 4.3.3.1) and (b) electro-migration (see section 4.3.3.2) parts.

4.3.3.2 Electro-migration

Computation of the electro-migration term involves both the concentration, stored at cell centers, and the potential, stored at cell vertices. Starting from the differential form (4.9) and applying the divergence theorem, we obtain:

$$\int_{\mathcal{C}_c} \nabla \cdot (c\,\nabla\psi)\;\mathrm{d}\mathcal{A} = \int_{\partial\mathcal{C}_c} \hat{\mathbf{n}} \cdot \nabla\psi\; c\,\mathrm{d}\ell.$$

Again, let us consider the discretization of the flux term at the right boundary, x^+ (see figure 4.2(b)). Similar to equation (4.13), we write:

$$\int_{\partial\mathcal{C}_c} \hat{\mathbf{n}} \cdot \nabla\psi\; c\,\mathrm{d}\ell \approx \overline{F^e}\Delta y_c. \tag{4.18}$$

Here, $\overline{F^e}$ is the average electro-migrative flux, where the averaging is defined as in (4.14) and individual fluxes are written as:

$$F^e_j = \frac{1}{2}\left(\widetilde{\psi}_x|_{v_j} + \widetilde{\psi}_x|_{v_{j+1}}\right)\; c_{j+1/2}, \tag{4.19}$$

where $\widetilde{\psi}_x$ is the second-order nodal approximation to $\partial\psi/\partial x$ as described in [101, 35, 60], $c_{j+1/2}$ is a second-order accurate approximation of the concentration field evaluated at $(x_c + \Delta x_c/2, y_j)$. The simplest approach for obtaining such a second-order accurate result involves linear interpolation of concentration values, which can be done by considering a triangulation of cell centers. Moreover, such an interpolation scheme is also necessary when solving the Poisson equation. We therefore postpone the full description of such interpolation scheme to section

4.3.3.3. Combining (4.18) and (4.19), we obtain the final form of the discretization:

$$\int_{\partial \mathcal{C}_c} \hat{\mathbf{n}} \cdot \nabla \psi \, c \, \mathrm{d}\ell \approx \frac{1}{2} \sum_{j=1}^{N_{x+}} \left(\widetilde{\psi}_x|_{v_j} + \widetilde{\psi}_x|_{v_{j+1}} \right) \, c_{j+1/2} \, \Delta y_j. \tag{4.20}$$

Just as before, application of discretization (4.20) to remaining cell boundaries is trivial. We also note that the extension to three spatial dimensions, though technically more involved, follows the same logic as presented here.

4.3.3.3 Cell data interpolation

As discussed earlier, a second-order accurate, cell-based interpolation scheme is required in our method. This is needed when in computing $c_{j+1/2}$ in (4.19) and when evaluating the source term in the Poisson equation (4.11) at the nodes. Unlike node-based interpolations, cell-based interpolations are expensive to construct on quadtrees. This is because cell centers, unlike nodes, do not follow a particular logical coordination and generally a triangulation is necessary. As a result we use linear interpolation on a locally triangulated grid as illustrated in figure 4.3.

To construct the local triangulation, we merely generate a list of all neighboring cell centers in an arbitrary order (clock-wise or counter-clock-wise). Note that no effort is made to obtain "optimum" triangulation in any sense. Given an arbitrary

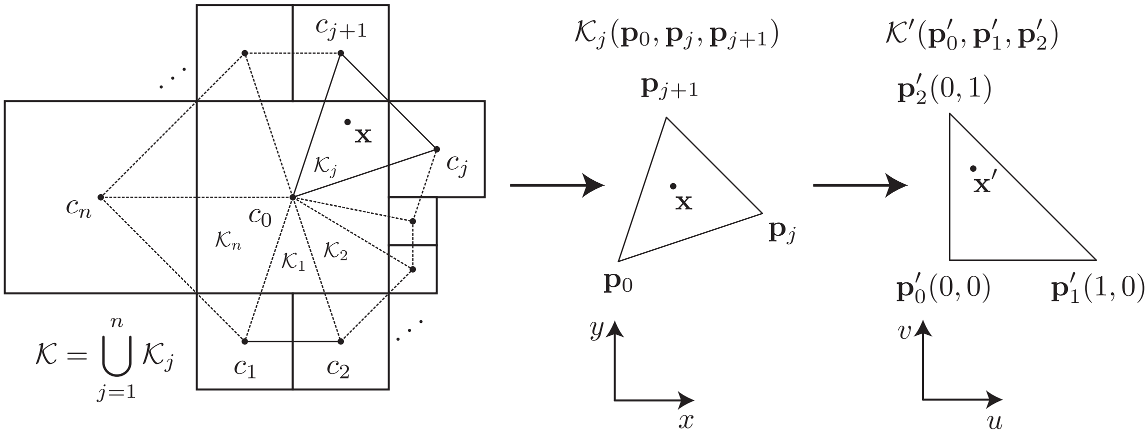


Figure 4.3: Cell data interpolation. To interpolate concentration fields from cell centers to any point $\mathbf{x}$, it is first needed to find the smallest cell $\mathcal{C}_0$ in the tree that contains the point $\mathbf{x}$. This can be done by recursively searching the quadtree data structure. Next, the triangulation $\mathcal{K} = \bigcup_{j=1}^{n} \mathcal{K}_j$ is constructed by including all neighboring cells in either clock-wise or counter-clock-wise order. Next, the correct triangle $\mathcal{K}_j$ is found by mapping all triangles to the reference triangle and testing if the point is encompassed by the triangle (see section 4.3.3.3). Finally, the concentration value at point $\mathbf{x}$ is approximated by linear interpolation on the reference triangle.

point $\mathbf{x}$ in the triangulation $\mathcal{K} = \bigcup_{j=1}^{n} \mathcal{K}_j$, where $\mathcal{K}_j$'s are triangles, it is first required to find the triangle $\mathcal{K}_j(\mathbf{p}_0, \mathbf{p}_j, \mathbf{p}_{j+1})$ such that $\mathbf{x} \in \mathcal{K}_j$. This can easily be done by mapping the triangle $\mathcal{K}_j$ to the reference triangle in the (u, v) domain (see figure 4.3). The mapping is given by:

$$\begin{pmatrix} x_i - x_0 & x_{j+1} - x_0 \\ y_i - y_0 & y_{j+1} - y_0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix},$$

which results in:

$$\begin{aligned} u &= \frac{\mathbf{r} \times \mathbf{r}_{j+1}}{\mathbf{r}_j \times \mathbf{r}_{j+1}}, \\ v &= \frac{\mathbf{r} \times \mathbf{r}_j}{\mathbf{r}_j \times \mathbf{r}_{j+1}}. \end{aligned}$$

Here, $\mathbf{r} = \mathbf{x} - \mathbf{p}_0$, $\mathbf{r}_j = \mathbf{p}_j - \mathbf{p}_0$, $\mathbf{r}_{j+1} = \mathbf{p}_{j+1} - \mathbf{p}_0$, and '×' denotes the cross product. In the (u, v) domain, point $\mathbf{x}$ is inside the triangle $\mathcal{K}_j$ if and only if $u \geq 0$, and $v \geq 0$, and $u + v \leq 1$.

Mapping $\mathcal{K}_j$ to the standard triangle also makes the interpolation straightforward. Using linear basis functions, $N_0(u, v) = 1 - u - v$, $N_1(u, v) = u$, and $N_2(u, v) = v$, the interpolated value $c(\mathbf{x})$ is given by:

$$c(\mathbf{x}) = \sum_{i=0}^{2} c'_i N_i(u, v),$$

where $c'_0 = c_0$, $c'_1 = c_j$, and $c'_2 = c_{j+1}$.

4.3.3.4 Boundary conditions

To impose boundary conditions we use the cut-cell method [118, 60]. Consider cell $\mathcal{C}_c$ cut by the boundary Γ as illustrated in figure 4.4. We note that our grid generation scheme, based on the level-set method, ensures that the grid is locally uniform close to the boundary, which facilitates the discretization of boundary conditions. To impose the boundary conditions, let us consider only the cations and note that equation (4.5) may be written as:

$$\hat{\mathbf{n}} \cdot \mathbf{J} = \hat{\mathbf{n}} \cdot (\mathbf{J}_d + \mathbf{J}_e) = 0,$$

which includes both the diffusion and the electro-migration terms. It is then concluded that the total flux term $\tilde{\mathbf{J}}^{n+1}$ disappears on the electrode boundary, i.e. $\hat{\mathbf{n}} \cdot \tilde{\mathbf{J}}^{n+1}|_\Gamma = 0$. Thus, integrating the right-hand side of equation (4.9) in $\mathcal{C}_c \cap \Omega^-$ yields:

$$\int_{\mathcal{C}_c \cap \Omega^-} \nabla^2 c^{n+1} + \nabla \cdot (c^n \, \nabla \psi^n) \, \mathrm{d}\mathcal{A} = \int_{\partial \mathcal{C}_c \cap \Omega^-} \hat{\mathbf{n}} \cdot \left(\nabla c^{n+1} + c^n \nabla \psi^n \right) \, \mathrm{d}\ell,$$

where the divergence theorem is utilized. Since the grid is locally uniform, the discretizations of the diffusion and the electro-migrations terms are trivial: considering the right boundary, x^+, the diffusion part becomes:

$$\int_{x^+\cap\Omega^-} \hat{\mathbf{n}} \cdot \nabla c^{n+1}\, \mathrm{d}\ell = \frac{c_j^{n+1} - c_c^{n+1}}{\Delta x} \ell_{x^+}, \tag{4.21}$$

where $0 \leq l_{x^+} \leq \Delta y$ is the length of right side of cell $\mathcal{C}_c$ that is in Ω^- and is computed as:

$$\ell_{x^+} = \begin{cases} \Delta y & \text{if } \phi^{+-} < 0 \text{ and } \phi^{++} < 0 \\ \Delta y \left(\frac{\phi^{+-}}{\phi^{+-} - \phi^{++}} \right) & \text{if } \phi^{+-} < 0 \text{ and } \phi^{++} > 0 \\ \Delta y \left(\frac{\phi^{++}}{\phi^{++} - \phi^{+-}} \right) & \text{if } \phi^{+-} > 0 \text{ and } \phi^{++} < 0 \\ 0 & \text{if } \phi^{+-} > 0 \text{ and } \phi^{++} > 0 \end{cases}.$$

Similarly, the electro-migration term can be discretized as:

$$\int_{x^+\cap\Omega^-} \hat{\mathbf{n}} \cdot \nabla \psi^n c^n\, \mathrm{d}\ell = \frac{\widetilde{\psi_x^n}^{++} + \widetilde{\psi_x^n}^{+-}}{2} \frac{c_c^n + c_j^n}{2} \ell_{x^+}. \tag{4.22}$$

4.3.3.5 Geometric extrapolation

Accurate calculation of gradients, right at the boundary, requires data that resides in the Ω^+ domain, which may not be available since the solution is com-

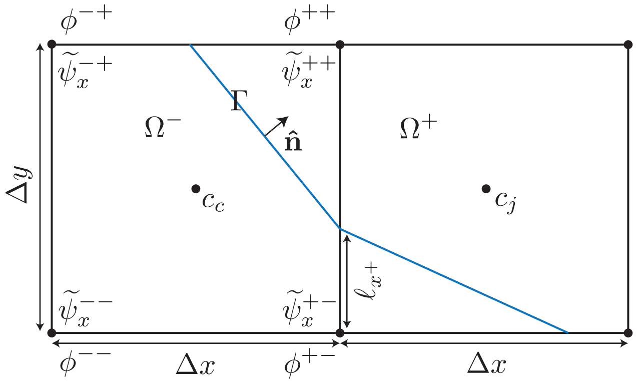


Figure 4.4: Schematic of a cell near the boundary. The boundary is illustrated as the blue solid line crossing the cell $\mathcal{C}_c$ and dividing it into Ω^+, Ω^-, and Γ subdomains. Note that our grid generation method as described in section 4.3.1 guarantees that the grid is locally uniform for all the cell crossed by Γ.

puted in the Ω^- domain. As a result it is necessary to be able to "extend" valid solutions from Ω^- into Ω^+. This can be achieved via extrapolation methods. In section 2.3.4 we described a PDE approach to the extrapolation problem following the works of [6]. While this was useful in the context of Poisson-Boltzmann solvers, the proposed method is computationally expensive when used at every single time-step of PNP calculations. We therefore propose an alternative, geometric approach to the extrapolation problem that is significantly faster. This methodology may not be as robust if the interface is under-resolved and therefore one must impose that the interface is well-resolved at all times.

Our new method is based on building and extrapolating polynomials in the normal direction to the interface. Figure 4.5 schematically depicts a situation where a quantity of interest (the potential field for example) needs to be extrapolated from Ω^- into Ω^+. The first step in constructing the polynomial involves projecting the point $\mathbf{x}_i$ onto the interface. The location of the projection point $\mathbf{x}_0$ can easily be calculated from the level-set function if it is a signed-distance function:

$$\mathbf{x}_0 = \mathbf{x}_i - \phi_i \hat{\mathbf{n}}, \tag{4.23}$$

where $\hat{\mathbf{n}} = -\frac{\nabla\phi}{|\nabla\phi|}$ is the unit normal vector on the interface, pointing toward Ω^+. At the leading order, a zeroth-order polynomial may be constructed by simply writing

$$F_i = F_0 = F_{\mathrm{D}}(\mathbf{x}_0),$$

where it is assumed that a Dirichlet boundary condition, F_{D}, is available for the quantity F. If such an information is not available or a higher order method is desired, more points $\mathbf{x}_j$ in the normal direction may be obtained according to

$$\mathbf{x}_j = \mathbf{x}_0 - \delta_j \hat{\mathbf{n}}, \quad j = 1, 2, \cdots \tag{4.24}$$

where δ_j is an arbitrary grid-dependent normal distance chosen for point $\mathbf{x}_j$ in Ω^-. Proper choice for δ_j, and thus $\mathbf{x}_j$, depends on several factors. First, δ_j should scale with the minimum grid size to guarantee convergence. Second, it is crucial that no two $\mathbf{x}_j$'s are very close, otherwise the constructed polynomial will suffer from very large coefficients that adversely affect the accuracy. Third, $\mathbf{x}_j$ points should be chosen such that it is relatively easy to compute the function values F_j.

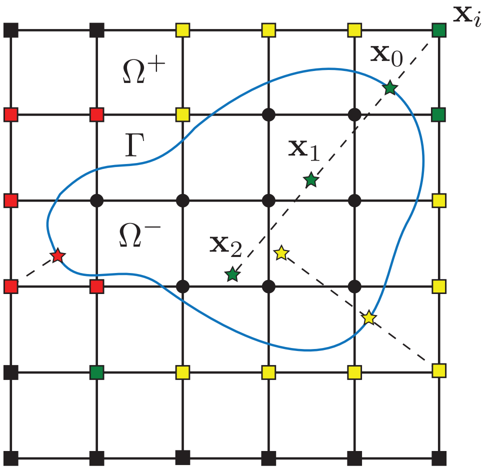


Figure 4.5: Geometric extrapolation method. To extrapolate a valid quantity F from Ω^- to point $\mathbf{x}_i$ in the Ω^+ domain, first the projection point, $\mathbf{x}_0$, is computed according to equation (4.23). Next, and depending on the desired order of accuracy, further points in the normal directions are computed according to (4.24). Note, however, that in regions of the domain where curvature of the boundary is larger than what can be captured by the grid, it is necessary to lower the degree of extrapolating function. This is done by removing points that will result in use of invalid data from Ω^+ domain. Here green squares represent points in the Ω^+ domain for which three valid points (including the projection point) may be computed which results in a quadratic extrapolating function. This is in contrast to yellow and red squares for which only two and one points are found, respectively. In these cases a linear and constant polynomials are used for extrapolation, respectively.

The simplest approach to chose the points $\mathbf{x}_j$ that satisfies all three conditions is to choose them equidistant to each other, i.e. $\delta_j = j\sqrt{\Delta x^2 + \Delta y^2}$. Furthermore, for such a point, F_j can simply be evaluated using a bilinear or biquadratic interpolating function $F_\mathrm{I}(\mathbf{x})$ [99]. One should note that these interpolating functions may not always be available. For instance if the interface is not smooth enough or accurately resolved by the grid, it is possible that one or more $\mathbf{x}_j$ points end up too close to the interface, or even cross the interface again and end up in Ω^+ (see figure 4.5). In both situations, the interpolating function F_I could potentially include points that belong to Ω^+ and have invalid values.

To remedy this problem, one either has to lower the degree of extrapolation and ignore such points or have a fall-back method, such as one-dimensional interpolation in x or y direction or resort to least square fits. Here, we simply lower the degree of extrapolating function and do not use such points in the final polynomial. In fact, Newton polynomials are built in a hierarchical fashion which makes this kind of approach easy to implement. Thus, we organize the final algorithm as follows: to construct a polynomial of maximum degree $\mathcal{M}$, that extrapolates a value F_i at point $\mathbf{x}_i$ in the Ω^+ domain from the valid quantity F in the Ω^- domain, do:

1. Compute the projected point $\mathbf{x}_0$ using equation (4.23).
2. If a Dirichlet boundary condition, F_D, is available for F, compute $F_0 = F_D(\mathbf{x}_0)$ and add $(\mathbf{x}_0, F_\mathrm{D})$ to the polynomial list. Otherwise, skip this step.
3. For $j = 1 : \mathcal{M}$, compute the point $\mathbf{x}_j$ according to equation (4.24). If $\mathbf{x}_j$ can be used for the extrapolation procedure, add $(\mathbf{x}_j, F_\mathrm{I}(\mathbf{x}_j))$ to the polynomial

list. If it cannot be used for the extrapolation procedure, terminate the loop. Here we use a biquadratic interpolating function for F_{I} for increased accuracy. However, bilinear methods are preferred if the interface is not very smooth or is under-resolved.

4. Construct the extrapolating polynomial F_{E} using as many points as possible using Newton polynomials:

$$F_{\mathrm{E}}(\eta) = \sum_{j=0}^{\mathcal{M}} [F_0, F_1, \dots, F_j] N_j(\eta),$$

where η denotes the coordinate in the normal direction and $[F_0, F_1, \dots, F_j]$ denotes the divided difference table. Finally $N_j(\eta)$ are the basis functions with $N_0(\eta) = 1$, and the remaining ones defined as:

$$N_j(\eta) = \prod_{k=0}^{j-1} (\eta - \eta_k).$$

5. The extrapolated value at $\mathbf{x}_i$ simply becomes $F_i = F_{\mathrm{E}}(\phi_i)$.

4.3.4 Notes on extension to three spatial dimensions

The algorithm presented here was primarily targeted toward two-dimensional problems. However, most parts of the algorithm can easily be extended to three spatial dimensions. Specifically, handling the Poisson equation (4.11) in three spatial dimensions is analogous to treatment of Poisson-Boltzmann equation which is explained in details in 2.3.2 and references therein. Also, the finite volume discretization of diffusion and electro-migration parts, as presented in sections 4.3.3.1 and 4.3.3.2, conceptually remain unchanged. The only required modification is to replace surface and line integrals to volume and surface integrals, respectively, and evaluate fluxes at the face center rather than edge center (c.f. [87] for similar approaches). For cells that are far away from the interface this modification is trivial. For cells that are cut by the interface, however, one is required to replace the length fraction, l_{x^+}, with an appropriate face fraction term. The computation of this face fraction, along with any possible integration on the interface, can be achieved using the geometric integration technique introduced in [100].

One ingredient of our algorithm that does not trivially extend to three spatial dimensions in the cell-based interpolation discussed in section 4.3.3.3. Note that although the general approach, i.e. linear interpolation based on triangulation of cell centers, is still viable, the way this triangulation is built in two dimensions does not extend to three dimensions. One way to solve this problem is to use a Delaunay triangulation algorithm that only involves the neighboring cell centers. Another way to approach this problem, which does not involve triangulation at all, is to construct a local, linear, least-squares interpolation that only involve neighboring cells. This is always possible since each cell has at least one neighboring cell in each of x, y, and z directions. Finally, it is easy to note that the geometric extrapolation technique introduced in section 4.3.3.5 trivially extends to three dimensions provided that a cell-based interpolation is available.

4.4 Accuracy analysis

In this section, we perform the accuracy analysis of our algorithm by conducting numerical experiments and computing the convergence rates. In addition, we consider two tests that illustrate the conservative properties of our method by measuring the change in the total mass in a closed system. To make the analysis more realistic, in all but one of tests we obtain the convergence rate through successive refinement and utilize the Richardson's extrapolation technique. Finally, a single "sanity-check" experiment is conducted where fictitious "exact solutions" are chosen and the grid is intensionally made to be non-graded in a random fashion to show the robustness of our algorithm.

Although calculation of convergence rate based on the Richardson's extrapolation technique is very well known, here we briefly describe it for the sake of completeness. Consider a quantity f that is accurate up to order r in space, i.e.

we have

$$f_h = f + Ch^r + o(h^r),$$

where f_h denotes a numerical approximation of the exact solution f on a grid G_h, h is a grid scale, for instance $h = h_{\min}$, and C is a continuous function that does not depend on h. Obtaining the solution on successively refined grids, $G_{h/2}$ and $G_{h/4}$, results in:

$$\begin{aligned} f_{h/2} &= f + C\frac{h^r}{2^r} + o(h^r), \\ f_{h/4} &= f + C\frac{h^r}{4^r} + o(h^r). \end{aligned}$$

As a result, the convergence rate can be estimated as

$$r \approx \log_2 \left(\frac{\|f_h - f_{h/2}\|_p}{\|f_{h/2} - f_{h/4}\|_p} \right), \tag{4.25}$$

where the difference in the numerical solutions can be measure in any norm L_p. Moreover, once the convergence rate r is calculated, it is possible to estimate the error in the same norm via extrapolation:

$$\|e_h\|_p = \|f_h - f\|_p \approx \| \frac{2^r}{2^r - 1} \left(f_h - f_{h/2} \right) \|_p. \tag{4.26}$$

4.4.1 Conservativeness

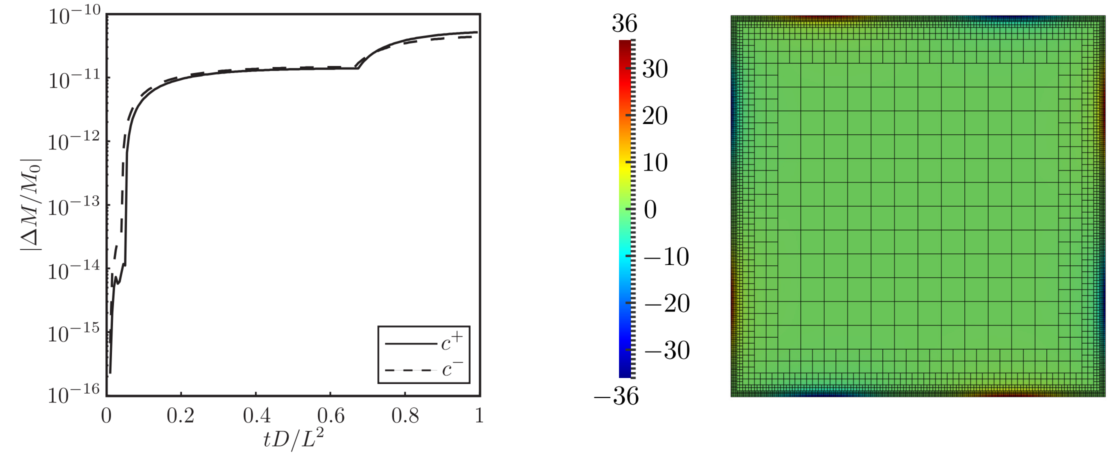
4.4.1.1 Two-dimensional cavity

For the first test, let us consider a two-dimensional domain $\Omega = [-1,1]\times[-1,1]$. No interface is present here meaning that the computational domain extends to the entire domain. A thin EDL case is considered for which $\kappa = 10$ and the potential on the walls is set to:

$$v(x,y) = \begin{cases} 5\sin(\pi y) & x = -1 \text{ and } -1 \leq y \leq 1 \\ -5\sin(\pi y) & x = 1 \text{ and } -1 \leq y \leq 1 \\ -5\sin(\pi x) & y = -1 \text{ and } -1 \leq x \leq 1 \\ 5\sin(\pi x) & y = 1 \text{ and } -1 \leq x \leq 1 \end{cases}$$

Finally, a grid resolution of $(l_{\max} = 8, l_{\min} = 4)$ and a constant time-step of $\Delta t = 5 \times 10^{-3}$ is chosen for this calculation. Figure 4.6(b) illustrates a snapshot of the charge density distribution at final time along with the adaptive grid in the domain.

To measure conservativeness, the simulation is initiated at $t = 0$, with uniform concentration fields $c_{\pm} = 0.5$, and integrated up to $t = 1$. At each time, the total



(a)

(b)

Figure 4.6: Results for the numerical example of section 4.4.1.1. (a) Change in the total mass of cations (solid) and anions (dashed) in a closed system. This graph clearly shows that our scheme is conservative. Small change, $\mathcal{O}(10^{-10})$, in mass is due to using an iterative linear solver for the diffusion part and round-off errors. (b) A snapshot of the charge density along with the adaptive grid used. The charge density is very large near the wall and quickly decays to zero in the bulk of the domain. This clearly illustrates the usefulness of adaptive grids in capturing large gradients in the EDL.

mass in the system is calculated via:

$$M_\pm = \int_\Omega c_\pm(\mathbf{x}, t) \, \mathrm{d}\Omega,$$

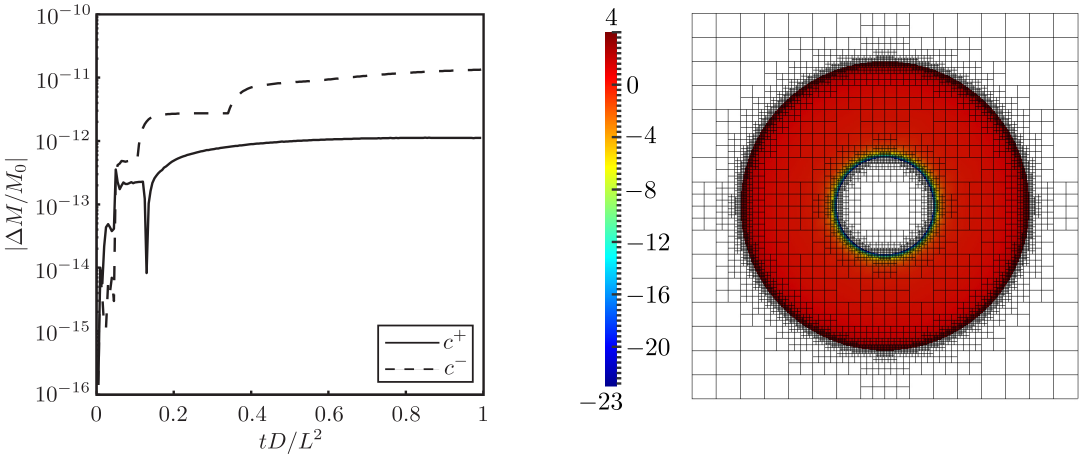
and the relative change in the mass is used as a metric for the conservativeness of the method. Since this is a closed system, the total mass in the system should not change in time. Indeed figure 4.6(a) illustrates that the method does conserve mass.

4.4.1.2 Concentric circular electrodes

To show that our method is conservative even when irregular geometries are present, again consider a domain $\Omega = [-1, 1] \times [-1, 1]$ and a level-set function defined as:

$$\begin{aligned}
\phi(x, y) &= \max(\phi_1, \phi_2), \\
\phi_1(x, y) &= 0.25 - \sqrt{x^2 + y^2}, \\
\phi_2(x, y) &= \sqrt{x^2 + y^2} - 0.75,
\end{aligned}$$

which defines Ω^- as the space between two concentric circles. All of the simulation constants are kept the same as their values in the previous test. The only exception is for the potential which is kept at 5 and -5 thermal volts on the inner and out circles, respectively. Figure 4.7(a) illustrates that the method remains conservative even in the presence of complex geometry. Figure 4.7(b) illustrates a snapshot of the solution along with the adaptive grid at final time.



(a)

(b)

Figure 4.7: Results for the numerical example of section 4.4.1.2. (a) Change in the total mass of cations (solid) and anions (dashed) in a closed domain defined by the Ω^-. Just as previous test, this graph illustrates the conservative properties of our algorithm. Again, small change, $\mathcal{O}(10^{-11})$, in mass is due to using iterative linear solver for the diffusion part and round-off errors. (b) snapshot of the charge density along with the adaptive grid used. Charge density is very large near the inner circle and quickly decays to zero in the bulk. Note that EDL forms at both electrodes but gradients are stronger on the inner circle due to stronger electric fields.

4.4.2 Spatial convergence rate

In this section we consider a series of different geometries and solve the PNP equations using our algorithm. In each case, a base grid is chosen and during successive refinement, every cell is split to ensure uniform convergence in the domain. Moreover, to ensure that the error due to time discretization is also reduced at the same rate, time-step is reduced such that $\Delta t \propto h_{\text{min}}^2$ at each refinement step. Note that this is done to ensure uniform decay of temporal errors and should not be confused with a stability criterion (c.f. section 4.4.3).

On each grid the PNP equations are integrated up to a final time after which equations (4.25) and (4.26) are utilized to measure the convergence rate and error in space. Since the calculation of each convergence rate in (4.25) requires 3 consecutive refinements, a total of 5 refinement steps are chosen. Once consecutive error terms are computed using (4.26), the overall convergence rates are computed via fitting the following curve to the error:

$$\|e_h\|_p = Ch_{\text{min}}^r. \tag{4.27}$$

Note that here we are mainly concerned with the convergence in space. Further discussion of choosing a proper time-step and convergence in time is provided in section 4.4.3

4.4.2.1 Flat electrodes

As the first example, we consider a domain $\Omega = [-1.25, 1.25] \times [-1.25, 1.25]$, where the electrode's boundary is given via the following level-set function:

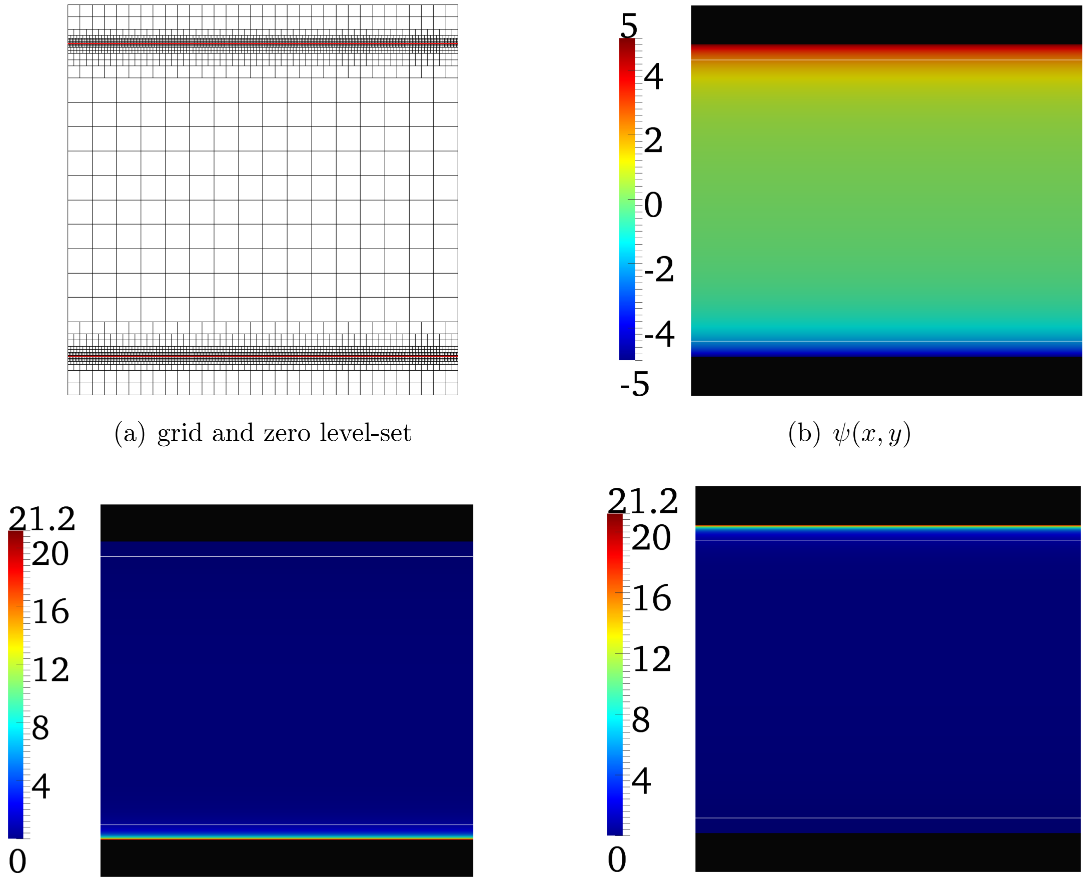
$$\phi(x, y) = |y| - 1.$$

A base resolution of $(l_{\max} = 8, l_{\min} = 4)$ is chosen for the grid with a coarse time-step of $\Delta t_c = 5 \times 10^{-3}$. The electric potential on the electrodes is set to 5 and -5 thermal volts for the top and bottom walls, respectively. Also a thin EDL limit of $\kappa^{-1} = 0.1$ is considered. Finally, periodic boundary conditions are chosen for all quantities in the x-direction. Figure 4.8 illustrates both the base grid and the solution at the final time of $t_f = 100 \times \Delta t_c = 0.05$ on the finest grid. Figure 4.9 gives the convergence rate in the L_1, L_2 and L_∞ norms under grid refinement. Notice that the number in parentheses is the overall convergence rate determined by fitting equation (4.27).

4.4.2.2 Circular electrode

Next we consider the domain $\Omega = [-1, 1] \times [-1, 1]$ along with the boundary Γ defined as a circle via

$$\phi(x, y) = 0.3 - \sqrt{x^2 + y^2}.$$



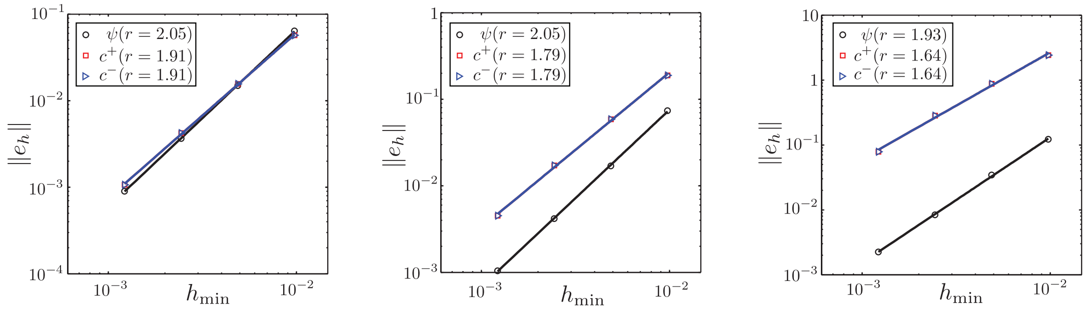
(a) grid and zero level-set

(b) $\psi(x, y)$

(c) $c^+(x, y)$

(d) $c^-(x, y)$

Figure 4.8: Solution of the PNP equations at the final time of $t_f = 100 \times \Delta t_c = 0.05$ for test 4.4.2.1. The base grid along with the zero level-set is shown in (a). The numerical solutions, (b-d), belong to the finest grid using the finest time-step. The EDL is marked with a white line representing the $\phi = -\kappa^{-1} = -0.1$ level-set, while the black shaded area represents the electrode (Ω^+ domain).



(a) L_1 norm

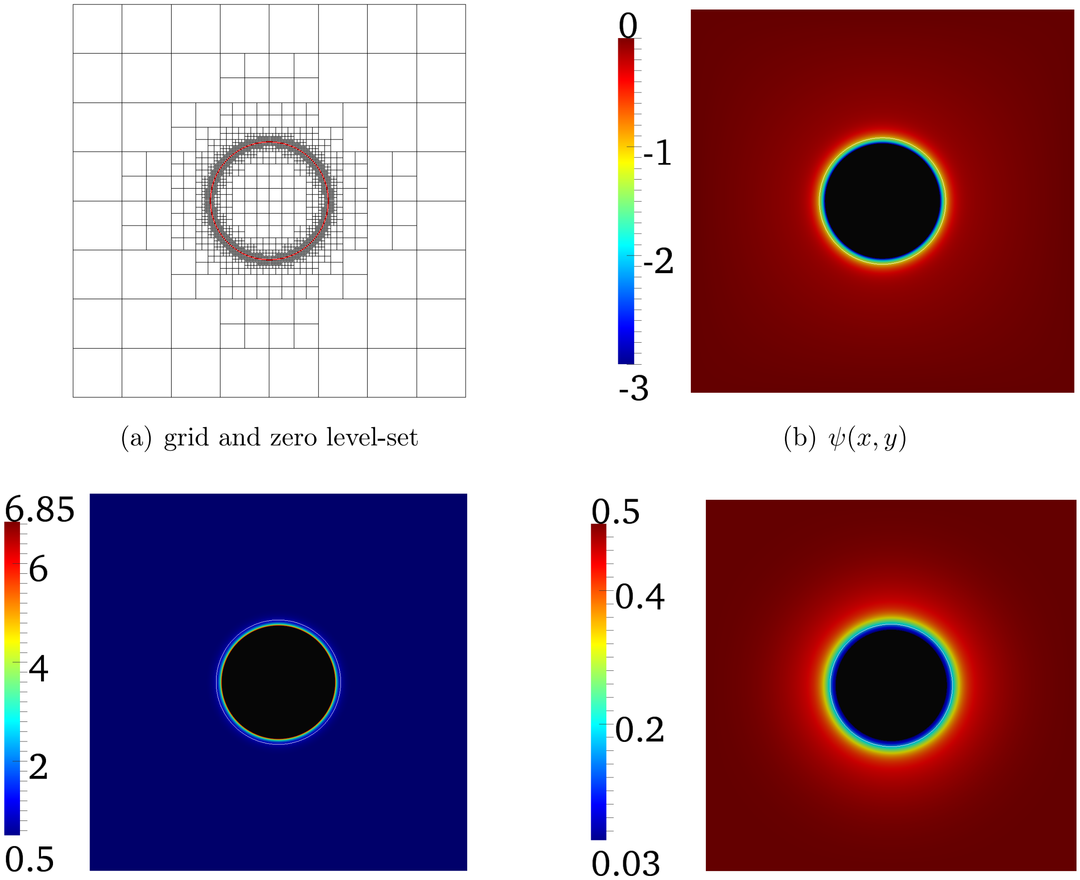
(b) L_2 norm

(c) L_∞ norm

Figure 4.9: Convergence rate of our algorithm for 4.4.2.1, obtained via Richardson extrapolation method. The numbers in parenthesis denote the overall rate obtained via fitting equation (4.27).

The potential is set to -3 thermal volts on the boundary and zero at the box boundaries of the domain, $\partial\Omega$. Ion concentrations satisfy the no-flux boundary condition on the electrode and the "bulk" condition at the box boundary, i.e. $c_{\pm}(x, y) = 0.5$ for $(x, y) \in \partial\Omega$. Finally an EDL thickness of $\kappa^{-1} = 0.03$ is chosen for this problem. Note that these numbers are chosen simply because they are physically relevant in studying supercapacitors.

A base grid of $(l_{\max} = 8, l_{\min} = 3)$ along with a coarse time-step of $\Delta t_c = 4.5 \times 10^{-4}$ are chosen. Figure 4.10 illustrates the initial grid and level-set along with the final solution on the finest grid while, figure 4.11 demonstrate that our method is second-order in all three norms for this numerical example.



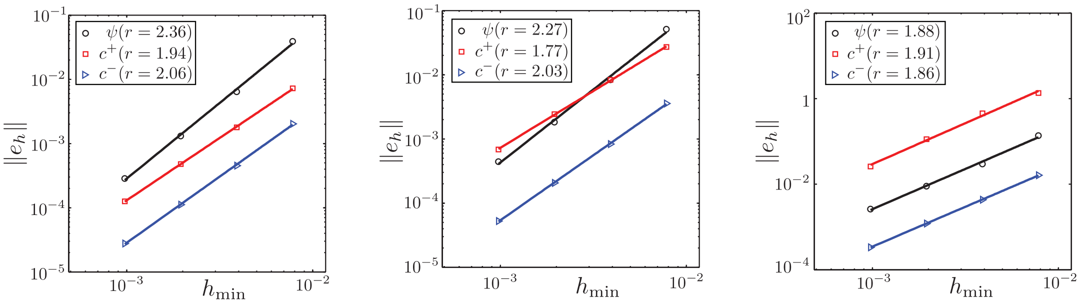
(a) grid and zero level-set

(b) $\psi(x, y)$

(c) $c^+(x, y)$

(d) $c^-(x, y)$

Figure 4.10: Solution of PNP for test 4.4.2.2 at the final time of $t_f = 100 \times \Delta t_c = 0.045$. The base grid along with the zero level-set is shown in (a). The numerical solutions, (b-d), are obtained on the finest grid using the finest time-step. The EDL is marked with a white circle, representing the $\phi = -\kappa^{-1} = -0.03$ contour. The black shaded area represents the electrode (Ω^+ domain).



(a) L_1 norm (b) L_2 norm (c) L_∞ norm

Figure 4.11: Convergence rate for test 4.4.2.2 using the Richardson extrapolation method. The numbers in parenthesis denote the overall rate obtained via fitting equation (4.27).

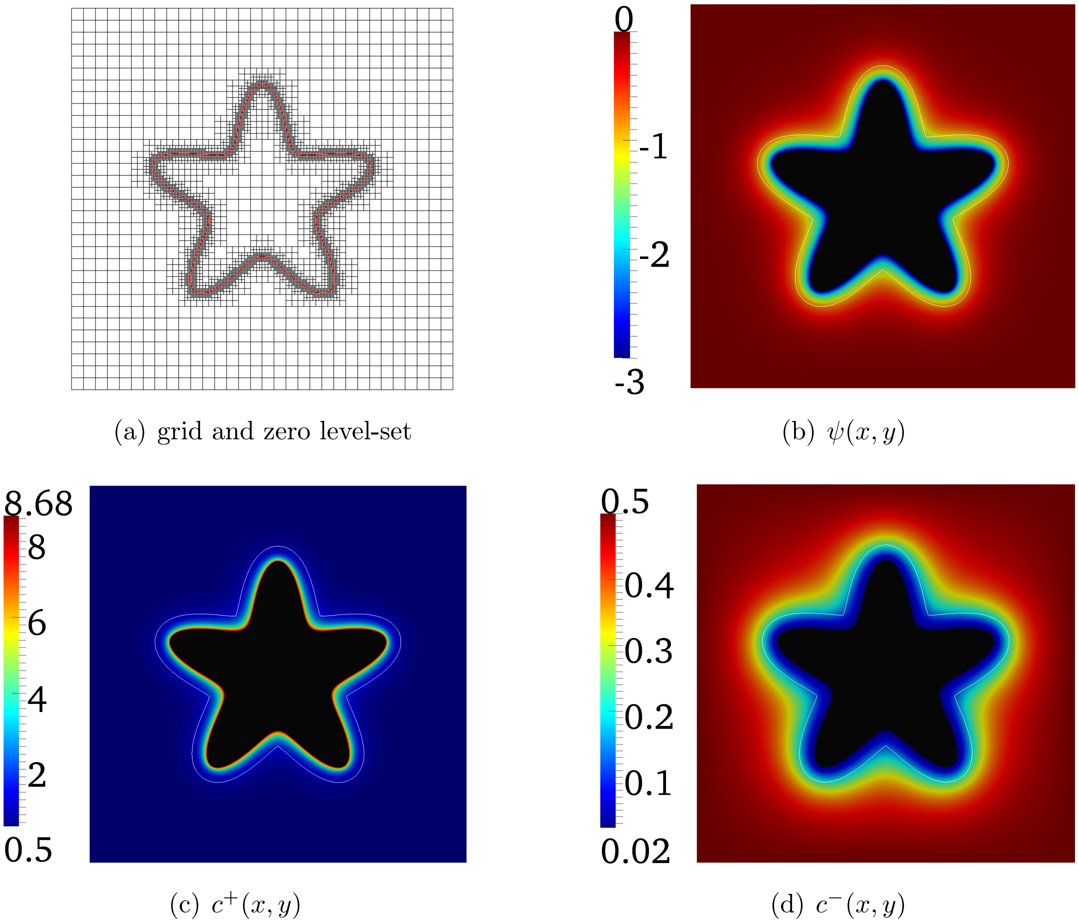
4.4.2.3 An irregular electrode

Next, let us consider the domain $\Omega = [-1.25, 1.25] \times [-1.25, 1.25]$ where the boundary is given by the zero level-set of a more complicated function:

$$\phi(x, y) = 0.5 - \sqrt{x^2 + y^2} + \frac{y^5 + 5x^4y - 10x^2y^3}{3(x^2 + y^2)^2}.$$

Similar boundary conditions are imposed for the potential and ion concentrations as in the previous test whereas a larger EDL with thickness of $\kappa^{-1} = 0.1$ is chosen. As for numerical parameters, a base grid of $(l_{\max} = 8, l_{\min} = 5)$ along with a coarse time-step of $\Delta t_c = 5 \times 10^{-3}$ are chosen. Figure 4.12 illustrates the initial grid and the zero level-set along with the final solutions on the finest grid. Again, figure

4.13 demonstrates that our algorithm is second order in all three norms for this numerical example.



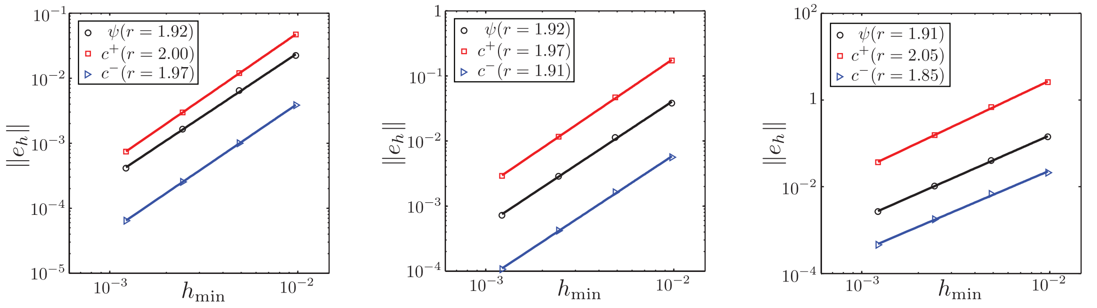
(a) grid and zero level-set

(b) $\psi(x, y)$

(c) $c^+(x, y)$

(d) $c^-(x, y)$

Figure 4.12: Solution of the PNP for test 4.4.2.3 at the final time of $t_f = 100 \times \Delta t_c = 0.05$. The base grid along with the zero level-set is shown in (a). The numerical solutions, (b-d), are computed on the finest grid using the finest time-step. The EDL is marked with a white curve, representing the $\phi = -\kappa^{-1} = -0.1$ contour. The black shaded area represents the electrode (Ω^+ domain).



(a) L_1 norm (b) L_2 norm (c) L_∞ norm

Figure 4.13: Convergence rate for test 4.4.2.3 using Richardson extrapolation method. The numbers in parenthesis denote the overall rate obtained via fitting equation (4.27).

4.4.2.4 Non-smooth electrode

Finally we want to make a comment about using our algorithm for non-smooth geometries. Consider the case when $\Omega = [-1.5, 1.5] \times [-1.5, 1.5]$ and the level-set function given by:

$$\phi(x, y) = \min(\phi_1, \phi_2),$$

where, ϕ_1 is given by:

$$\begin{aligned}
\phi_1(x, y) &= \min(f_1, f_2), \\
f_1(x, y) &= \sqrt{2|x| + |y|} - 1, \\
f_2(x, y) &= \sqrt{2|y| + |x|} - 1,
\end{aligned}$$

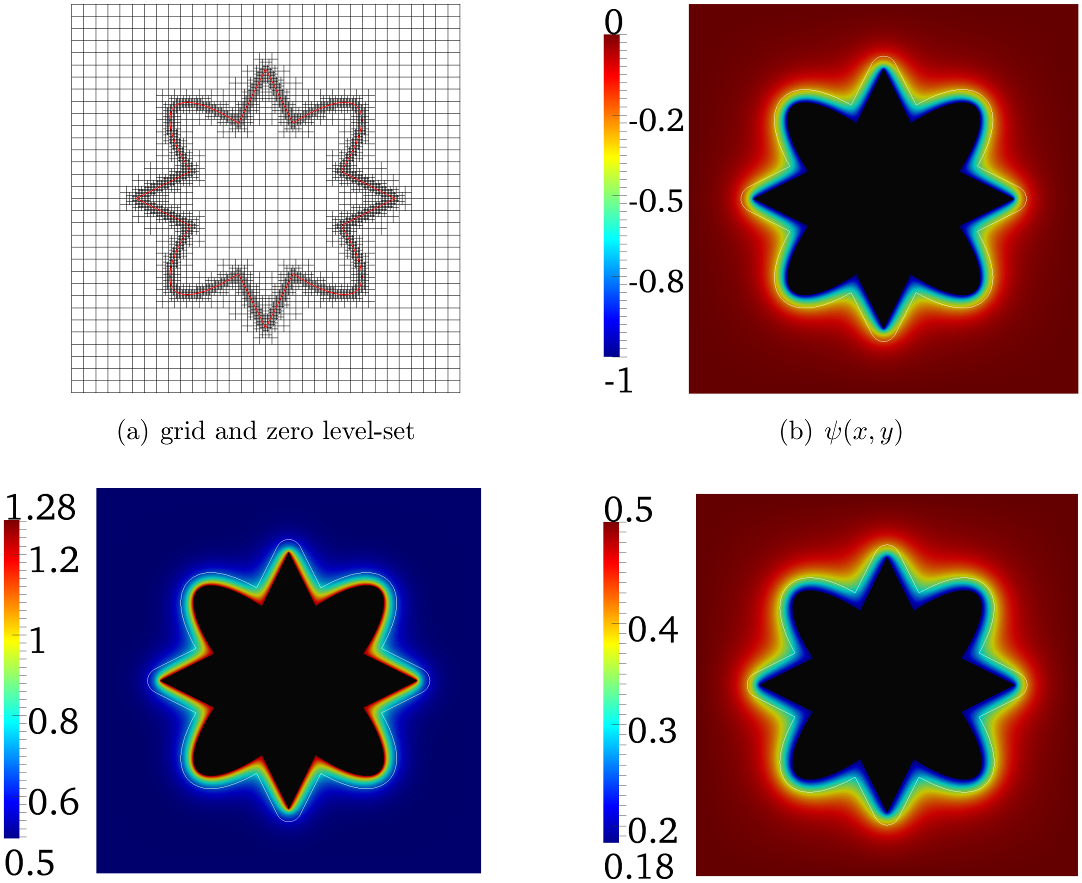
and ϕ_2 is given by a 45 degree rotation about the origin of ϕ_2':

$$\begin{aligned}
\phi_2'(x,y) &= \min(f_3, f_4), \\
f_3(x,y) &= \sqrt{10x^2 + y^2} - 1, \\
f_4(x,y) &= \sqrt{10y^2 + x^2} - 1.
\end{aligned}$$

The resulting boundary is shown in figure 4.14 with a $(l_{\text{max}} = 8, l_{\text{min}} = 5)$ adaptive grid. Here we keep all the parameters unchanged from the previous test with the exception of potential on the electrode which is set to -1 thermal volts. The final solution at $t_f = 100 \times \Delta t_c = 0.05$ on the finest grid is shown in figure 4.14.

Figure 4.15 illustrates the convergence rate analysis of our method. Unfortunately, this boundary contains a singularity which cannot be resolved by grid refinement. As a result, no matter how much the grid is refined, there will always be points close to the singularity for which one has to reduce extrapolating function to a constant polynomial. As a result, convergence rate in L_∞ is dropped by at least one order. Convergence rate in L_1 and L_2, however, is not affected by the singularity, meaning that loss of accuracy is purely a local property.

These results are typical of parabolic-type problems with geometrical singularities in the domain. Existing methods have been proposed to overcome this



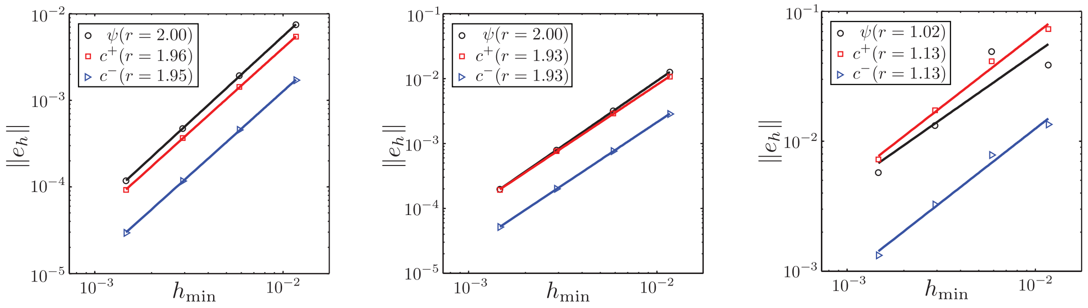
(a) grid and zero level-set

(b) $\psi(x, y)$

(c) $c^+(x, y)$

(d) $c^-(x, y)$

Figure 4.14: Solution of PNP for test 4.4.2.4 at the final time of $t_f = 100 \times \Delta t_c = 0.05$. Base grid along with the zero level-set is shown in (a). The numerical solutions, (b-d), belong to the finest grid using the finest time-step. EDL is marked with a white curve, representing $\phi = -\kappa^{-1} = -0.1$. The black shaded area represents the electrode (Ω^+ domain).



(a) L_1 norm

(b) L_2 norm

(c) L_∞ norm

Figure 4.15: Convergence rate of test 4.4.2.4 problem using Richardson extrapolation method. The number in parenthesis denotes the overall rate obtained via fitting equation (4.27). Notice how existence of singularities has dropped the convergence rate by at least one order in L_∞. This, however, is a local loss since L_1 and L_2 norms still show second-order convergence.

difficulty in the literature. For instance in [153] and [152] authors propose the so-called "Matched Interface and Boundary (MIB)" method for handling geometrical singularities for elliptic problems with applications to the Poisson-Boltzmann equation. In general it is possible to incorporate this method in our algorithm since the modifications only affect the boundary conditions where the grid is locally uniform. Nonetheless, for most practical applications, such as studying the charging kinetics of supercapacitors, the geometry does not involve geometrical singularities. For these problems the maximum curvature in the geometry, albeit maybe large, is finite and is eventually resolved on a fine enough grid. Although this may be prohibitively expensive on a uniform grid, our use of quadtree adaptive grids can dramatically reduce the computational cost in these cases.

4.4.3 Analytic test

Here we consider an example that will serve both as a test to show the robustness of our algorithms when applied to non-graded grids and also to demonstrate convergence both in space and time using a fictitious "exact solution". For this purpose, we consider the domain $\Omega = [-5, 5] \times [-5, 5]$ where the boundary is a circle represented as zero levelset of $\phi(x, y) = 1 - \sqrt{x^2 + y^2}$. The exact solutions are defined as:

$$\begin{aligned}
\psi(x, y, t) &= \exp(\kappa\, \phi(x, y))\,(1 - \exp(-t))\,, \\
c^+(x, y, t) &= \frac{1}{2}\exp\left(-\psi(x, y, t)\right), \\
c^-(x, y, t) &= \frac{1}{2}\exp\left(\psi(x, y, t)\right),
\end{aligned}$$

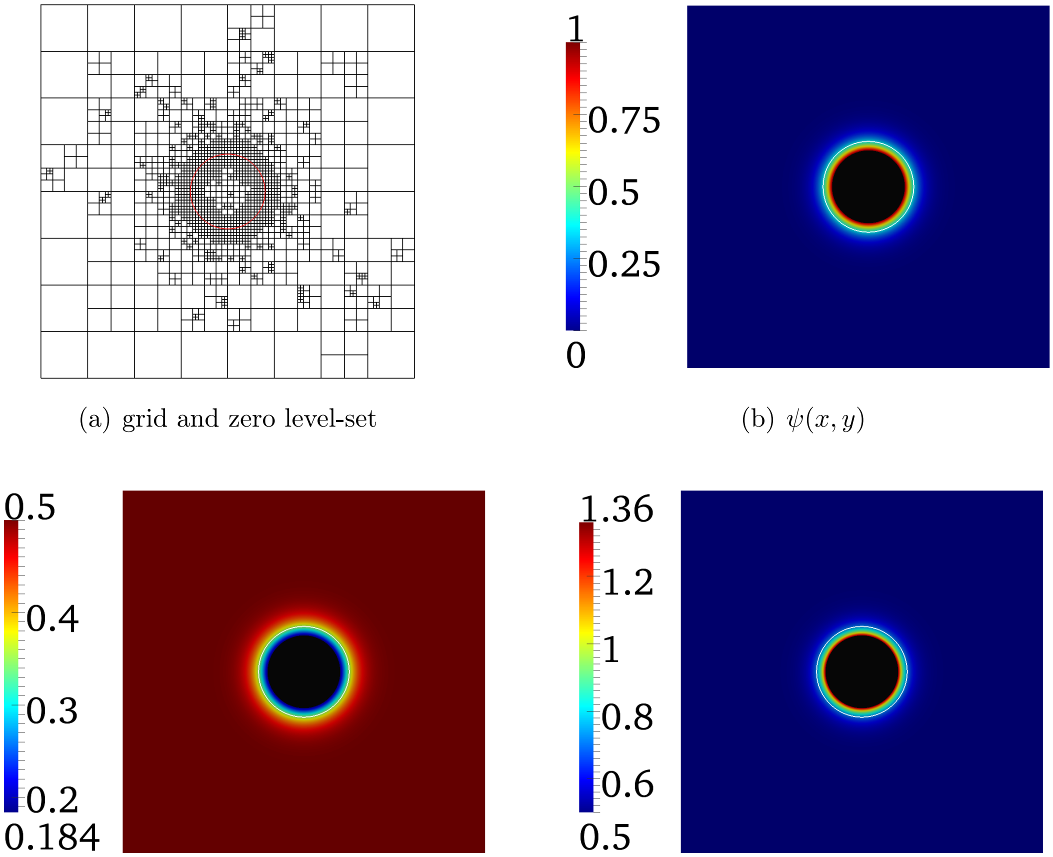
in the Ω^- sub-domain. It is easy to verify that these exact solutions satisfy the following PNP equations:

$$\begin{aligned}
\partial_t c_+ &= \nabla^2 c_+ + \nabla \cdot c_+ \nabla\psi + f(x, y, t), \\
\partial_t c_- &= \nabla^2 c_- - \nabla \cdot c_- \nabla\psi + g(x, y, t), \\
-\nabla^2\psi &= \kappa^2\,(c_+ - c_-) + h(x, y, t),
\end{aligned}$$

with the source terms f, g, and h obtained by plugging the exact solutions into the PNP equations. Moreover, these exact solutions are also utilized to obtain appropriate Dirichlet and no-flux boundary conditions on the electrode boundary, Γ, and domain boundary, $\partial\Omega$ (see equations (4.5), (4.5), and (4.6)).

To demonstrate the robustness of our algorithm, we initially apply the regular grid generation scheme discussed in sections 4.3.1 and 2.3.1 using a base resolution of ($l_{max} = 7$, $l_{min} = 3$). Next, the tree is randomly refined to ensure that the resulting grid is non-graded. This is illustrated in figure 4.16(a). Finally, the EDL thickness is set to $\kappa^{-1} = 0.25$ and an initial time-step of $\Delta t = 3.125 \times 10^{-2}$ is chosen for the integration. To measure the accuracy and convergence rate in time, the PNP equations are integrated up to $t = 0.125$. At each refinement step, the time-step is decreased by a factor of two and all cells are split. This ensures that temporal errors remain dominant during refinement. Once the errors are calculated, the overall temporal convergence rate, s, is computed by fitting $\|e_h\|_p = C\Delta t^s$ to the error. As expected, figure 4.17 illustrates first order convergence rate for all variables in L_1, L_2, and L_∞ norms.

Finally, to study the convergence rate in space, the PNP equations are integrated up to $t = 10$ using a fixed time-step of $\Delta t = 3.125 \times 10^{-2}$. Since the solution involves an exponential decay in time, temporal errors are minimal at $t = 10$ and the main contribution to the error is from spatial discretization. Fig-



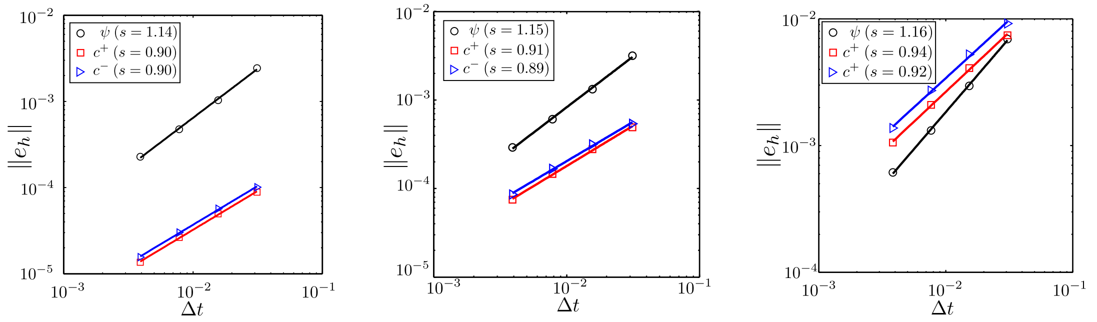
(a) grid and zero level-set

(b) $\psi(x, y)$

(c) $c^+(x, y)$

(d) $c^-(x, y)$

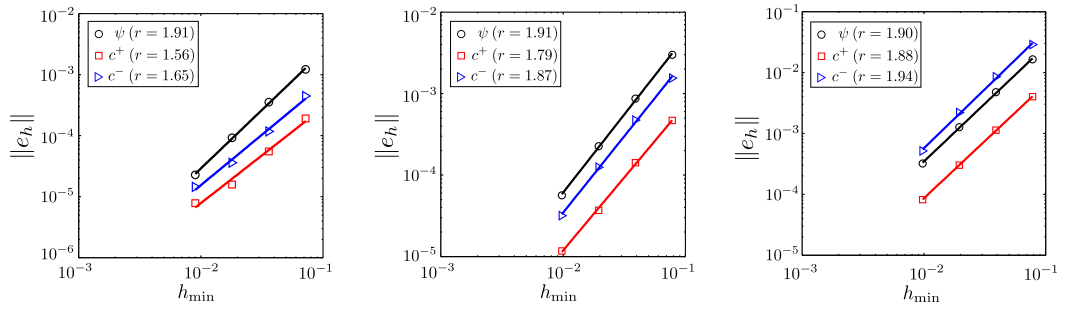
Figure 4.16: Solution of PNP for test 4.4.3. Base grid along with the zero level-set is shown in (a). The numerical solutions, (b-d), belong to the finest grid at $t = 10$. EDL is marked with a white curve, representing $\phi = -\kappa^{-1} = -0.25$. The black shaded area represents the electrode (Ω^+ domain).



(a) L_1 norm (b) L_2 norm (c) L_∞ norm

Figure 4.17: Temporal convergence rate of our algorithm for 4.4.3. The numbers in parenthesis denote the overall rates.

ure 4.18 illustrates that our algorithm is second order accurate in space, even when the underlying grid involves large jumps in the level of neighboring cells. Here, the overall convergence rate in space is calculated by fitting equation (4.27) to the error as before.



(a) L_1 norm (b) L_2 norm (c) L_∞ norm

Figure 4.18: Spatial convergence rate of our algorithm for 4.4.3. The numbers in parenthesis denote the overall rates.

We finish this chapter by noting that the presented algorithm is conservative, first order accurate in time and second order accurate in space, even for non-graded grids. An important question that we have left unanswered here regards the stability of our algorithm. While theoretical stability analysis of the algorithm is of great interest, it has proven to be quite challenging due to nonlinear nature of the equations and grid adaptivity. Nonetheless, in our studies we have never experienced any time-step restrictions with respect to the grid size or the jump in the level of neighboring cells. These observations are consistent with those made in [89].

The only time-step restriction we have encountered so far, seems to solely depend on the EDL thickness, κ^{-1}, and the applied potential. In particular, we have found $\Delta t \leq \kappa^{-2}$ as a reliable upper bound at small potentials. This is due to explicit treatment of the nonlinear electro-migration term and is consistent with the fact that EDL relaxation time scales as $\tau = \mathcal{O}(\kappa^{-2})$ [16]. At higher potentials, a smaller time-step is typically required but we are unable to give any specific recommendation.

Chapter 5

Charging Kinetics of Porous Electrodes

5.1 Introduction

Porous electrodes play a central role in various electrochemical devices and technologies, including electro-catalysts [11], batteries [107], fuel cells [123], supercapacitors [75, 136, 45], as well as in emerging technologies like capacitive deionization (CDI) [143, 121, 120, 144], and 'blue' energy harvesting through capacitive mixing of fresh and salt water (CAPMIX) [26, 126, 125]. The large surface area per unit volume (or mass) inherent to porous electrodes relative to planar electrodes enhances the rates and magnitude of currents and capacitance that can be achieved. This large surface area, however, is only effective if electrolyte transport occurs quickly enough that all pores are accessible in the relevant time.

The classic “Transmission-Line” (TL) model of de Levie [51], described below, continues to see widespread use in predicting ion transport in porous electrodes. The TL model, however, is built upon linearized electrolyte theories that omit crucial physics at the $\sim$ 100mV - 1V potentials relevant for most technologies. A more advanced model, developed by Biesheuvel and Bazant (BB) [20], reveals ion transport to be slowed significantly at the higher potentials relevant for porous electrode processes, reducing the achievable power densities.

Testing the validity of either of these two models, however, would require the full numerical solution of the nonlinear, coupled Poisson-Nernst-Planck (PNP) equations, generally in complex, three-dimensional geometries. Very few direct numerical simulations (DNS) have been performed to study the charging kinetics of porous electrodes, with exceptions limited to low potentials or simple geometries.

Here we use the numerical algorithms that was described in the previous chapter to perform DNS studies of the nonlinear PNP equations, fully resolving transient electric double layers under strong potentials in complex geometries. We directly test both linear (TL) and nonlinear (BB) transmission-line models against the full ion transport dynamics. Curiously, our studies reveal systems whose charging times are *underpredicted* by TL, yet *overpredicted* by BB, even by orders of magnitude. Our DNS results reveal surface conduction (SC), not included in either model, to play the key role in accelerating the charging. We derive a new effective transmission-line model (SC) that incorporates surface conduction and captures the DNS results quantitatively. Significantly, the SC mechanism depends strongly on electrode morphology, unlike previous effective theories, giving an important new quantity to target in the design of high-performance electrochemical systems.

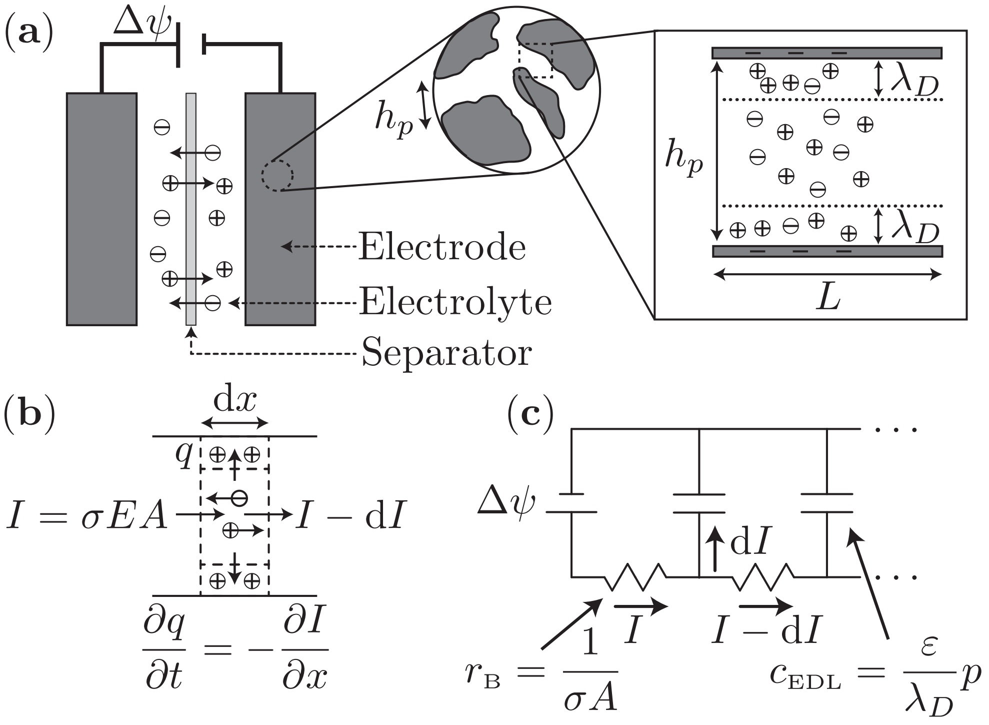


Figure 5.1: (a) Macroscopic porous electrodes derive high surface area from the many small pores they contain. The TL model assumes EDL thickness λ_D to be thin compared to the pore size h_p. (b) The charging current drives ions along the pore, some of which are diverted to unscreened electrode sections to form EDLs. Ion conservation gives the TL equation, (5.3). (c) The equivalent TL circuit uses linear resistors and capacitors to represent the bulk electrolyte and EDL, respectively.

5.2 Theory

Consider a model supercapacitor (fig. 5.1a), which stores energy within the electric double-layers (EDLs) that form around charged electrodes. The characteristic EDL thickness within symmetric, monovalent, binary electrolytes is given by $\lambda_D = \sqrt{\varepsilon k_\mathrm{B} T / 2 c_\infty e^2}$ [93], wherein c_∞ and e are the ions' number density and elementary charge, ε is the electrolyte permittivity, and $k_\mathrm{B}T$ is the thermal energy. At applied potentials lower than the thermal potential

$$\psi_\mathrm{th} = k_\mathrm{B} T / e, \tag{5.1}$$

EDLs act like linear, parallel plate capacitors, with specific capacitance (per unit area) $c_\mathrm{EDL} = \varepsilon / \lambda_D$ [93]. Unlike parallel plate capacitors, however, EDLs form conformally over non-planar electrodes. Therefore, combining the high surface area of porous electrodes ($\sim 1000\,\mathrm{m}^2/\mathrm{g}$ [136]) with the high capacitance per unit area ($\sim 10\,\mu\mathrm{F}/\mathrm{cm}^2$ [75]) of the EDL, yields ultra-high capacitance per mass. Furthermore, the absence of chemical reactions gives faster charge/discharge rates when compared to batteries, for high power density energy storage [75]. Decreasing pore size increases the surface area – and thus specific capacitance – of the electrodes, but at the expense of increased ion transport resistance (and thus lower power density).

To model the charging dynamics of porous electrodes, we first consider a single pore of perimeter p_p, cross-sectional area A_p, and axis $\hat{\mathbf{x}}$ whose walls have been raised to a potential ψ_e relative to the bulk electrolyte (fig. 5.1b). Counter-ions are driven into the pore (and co-ions out) to form EDLs when they encounter un-screened electrode surfaces. We assume the effective (hydraulic) radius of the pore,

$$h_p = A_p/p_p, \tag{5.2}$$

to be large relative to the EDL ($h_p \gg \lambda_D$), so that most of the pore is electroneutral. The TL model treats electrode charging in the long-wavelength limit, where variations in electric field $\mathbf{E}$ occur over length scales much longer than h_p, so that $\mathbf{E} \approx -\psi_x(x)\hat{\mathbf{x}}$. This field drives a current $\mathbf{I} = -\sigma\psi_x(x)A_p\hat{\mathbf{x}}$ along the pore, where $\sigma = 2c_\infty e^2 D/k_{\mathrm{B}}T$ with ion diffusivity D. Locally charging the EDL consumes some of this current, and conservation requires $\partial q/\partial t = -\partial I/\partial x$. Assuming constant conductivity σ and linear EDL capacitance $q(x) = c_{\mathrm{EDL}}p_p(\psi(x)-\psi_e)$ gives the TL equation,

$$\frac{\partial \psi}{\partial t} = \left(\frac{h_p}{\lambda_D}\right) D \frac{\partial^2 \psi}{\partial x^2}. \tag{5.3}$$

The EDL charging front thus propagates diffusively along the pore, with a charging time scale

$$\tau_{\mathrm{TL}} = \frac{\lambda_D}{h_p}\frac{L^2}{D}, \tag{5.4}$$

for a pore of length L.

While the TL model has been used to interpret experiments [122] and optimize electrode morphologies [54], its assumptions of low potentials and thin EDLs ($\Delta\psi \ll \psi_{\text{th}}$ and $\lambda_D/h_p \ll 1$) are often violated in practice [136]. Newman and Tiedemann [107] extended the TL model to include morphological effects via volume-averaging methods and an effective pore radius h_p. The BB model [20] predicts high-potential charging kinetics to be slowed dramatically due to (i) nonlinear EDL capacitance $c_{\text{EDL}}(\Delta\psi)$ [129] and (ii) salt depletion in the pores, which decreases the local electrolyte conductivity.

Both TL and BB models approximate the Poisson-Nernst-Planck (PNP) equations for dilute ion transport,

$$\frac{\partial c_\pm}{\partial t} = D\nabla^2 c_\pm \pm \mu e \nabla \cdot (c_\pm \nabla\psi), \tag{5.5}$$

$$-\nabla^2\psi = \frac{e(c_+ - c_-)}{\varepsilon}, \tag{5.6}$$

where $\mu = D/k_{\text{B}}T$ is ion mobility and $c_\pm$ are the ion number densities [93]. Direct tests of the TL and BB models requires numerical solutions of the full PNP equations, which becomes challenging at high potentials due to the extremely sharp gradients within thin EDLs. The TL model has been validated with the linearized PNP equations for straight pores [130], whereas nonlinear PNP studies of charging kinetics [84] were limited to low electrode potentials.

Here, using the algorithms described in the previous chapter, we perform DNS study to understand the effects of applied potential and pore morphology on the charging dynamics of porous electrodes, focusing on thin EDLs in order to directly test the TL and BB models. As a comparison metric, we define the *charging fraction*,

$$\eta(t) = \frac{1}{q_\infty} \int_{\text{pore}} (c_+(t,\mathbf{x}) - c_-(t,\mathbf{x}))\, \mathrm{d}A, \tag{5.7}$$

which expresses the charge $q(t)$ driven into the electrode as a fraction of the steady-state charge q_∞ that develops at a given potential [102]. The (linearized) TL can be solved exactly to yield $\eta(t) = 1 - \sum_{n=0}^{\infty} 2\lambda_n^{-2} \exp(-\lambda_n^2 t/\tau_{\text{TL}})$, where $\lambda_n = n\pi + \pi/2$.

5.3 Charging kinetics

Figure 5.2b shows the charging dynamics $\eta(t)$ computed for three pore morphologies (fig. 5.2a) at low applied potentials ($\Delta\psi = 0.1\psi_{\text{th}}$). When scaled by the naive ion diffusion time L^2/D, these appear to depend on electrode morphology. Properly scaling time by τ_{TL} (5.4), however, collapses all computations onto the

TL prediction, illustrating the TL's quantitative validity for general morphologies at low potentials.

We now examine higher applied potentials, where the TL is expected to fail. Fig. 5.3a reveals charging dynamics for a straight-walled pore at $\Delta\psi = 3$, 5, and 7.5 ψ_{th} to be up to an order of magnitude slower than TL predictions. Such slowing is predicted by the BB model, which accounts for both nonlinear EDL capacitance and salt depletion of the pore. Curiously, however, the BB model dramatically *overpredicts* the slowing (fig. 5.3b).

Detailed DNS results for the charging current (fig. 5.4) reveal a strong current within the highly-charged EDLs. Such excess surface currents, for example, are known to cause non-monotonic mobilities in electrophoresis [113]. This surface conduction [93] provides an additional charging pathway (5.3c) that enters the effective circuit diagram as a nonlinear resistor in parallel with the bulk resistor (fig. 5.3d). Notably, the resistance associated with the surface current is *reduced* as $\Delta\psi$ is increased [93], and ultimately 'short-circuits' the low-conductance bulk pores to decrease the charging time.

The BB model can be extended to include surface conduction for simple, straight-wall pores as in fig. 5.3. The BB model assumes an electro-neutral,

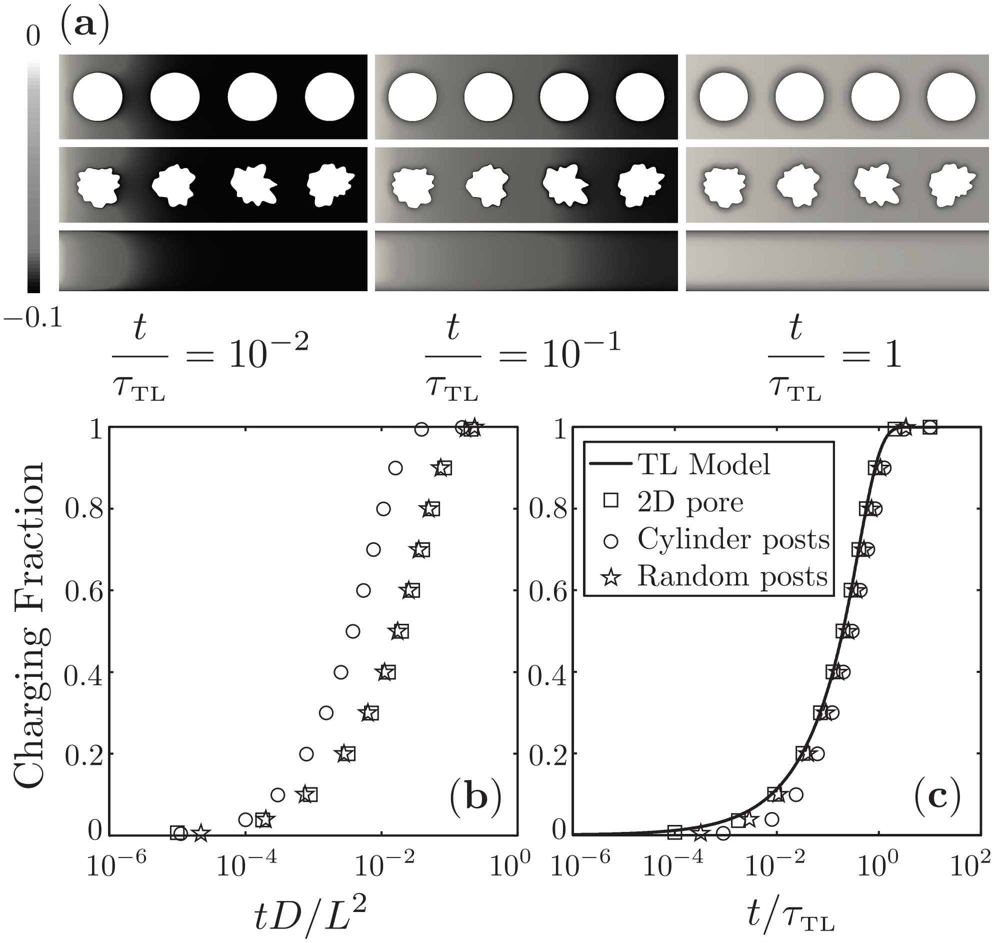


Figure 5.2: The computed charging fraction (5.7) of three model electrodes at $\Delta\psi = 0.1\psi_{\mathrm{th}}$. (a) The potential front at three different times looks remarkably similar for three different electrode morphologies. (b) Naively scaling time yields an apparent dependence on electrode morphology. (c) Scaling time by the TL time-scale (5.4) collapses all results onto the TL curve, confirming its validity at low potentials.

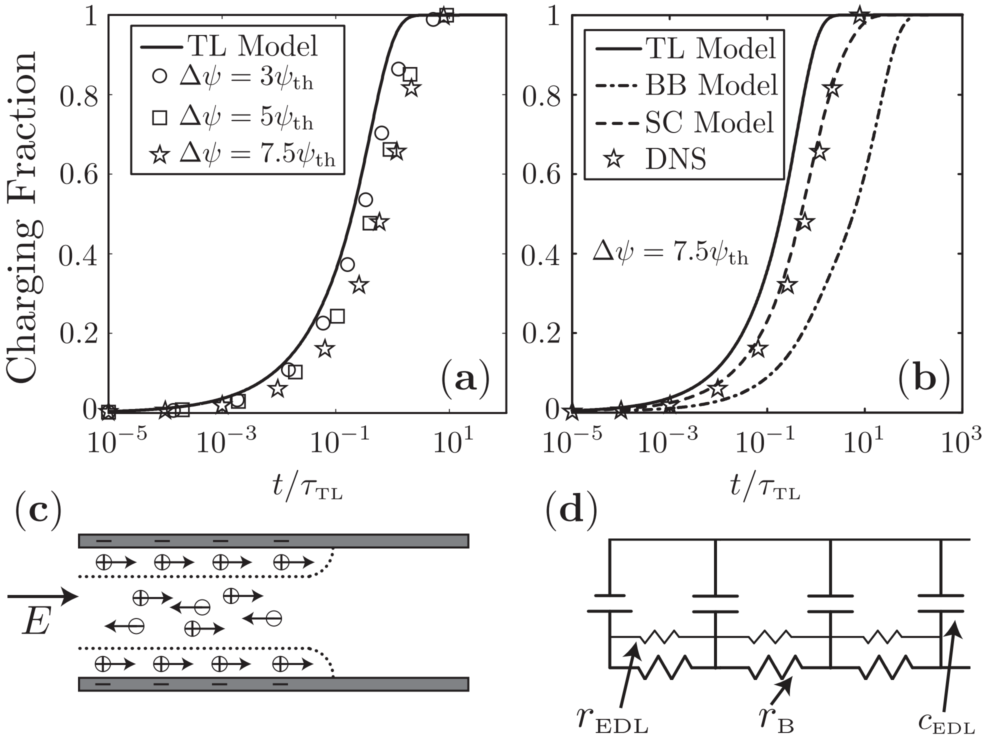


Figure 5.3: (a) The charging dynamics computed via DNS for a single straight pore at high potentials are slowed dramatically relative to the TL predictions. (b) Charging kinetics at $\Delta\psi = 7.5\psi_{\mathrm{th}}$ are underpredicted by TL, but *overpredicted* by BB. The SC model (5.12-5.13), which accounts for excess surface conduction within the EDL, quantitatively captures the charging dynamics. (c) The high-conductivity of EDLs at high potentials gives rise to an excess (surface) current. (d) Surface conduction adds a nonlinear resistor r_{EDL} in parallel to that of the bulk pore r_B, which effectively short-circuits the high-resistance bulk pores at high potentials to lower the charging time.

volume-averaged bulk, and enforces conservation of charge and salt [20]:

$$\frac{\partial}{\partial x}\left(c\frac{\partial \psi}{\partial x}\right) = \epsilon i, \tag{5.8}$$

$$\frac{\partial c}{\partial t} = \frac{\partial^2 c}{\partial x^2} - \epsilon j, \tag{5.9}$$

where $c(t,x)$ and $\psi(t,x)$ are area-averaged salt density and electric potential, i and j are the current and salt fluxes into the EDL, and $\epsilon = \lambda_D/h_p$. We now extend the BB model by including the surface currents in the conservation equations for the EDL,

$$\frac{\partial q}{\partial t} = \frac{\partial^2 q}{\partial x^2} + \frac{\partial}{\partial x}\left(w\frac{\partial \psi}{\partial x}\right) + i, \tag{5.10}$$

$$\frac{\partial w}{\partial t} = \frac{\partial^2 w}{\partial x^2} + \frac{\partial}{\partial x}\left(q\frac{\partial \psi}{\partial x}\right) + j, \tag{5.11}$$

where $q = q(c,\psi) = 2\sqrt{c}\sinh((\psi-\psi_e)/2)$ and $w = w(c,\psi) = 4\sqrt{c}\sinh^2((\psi-\psi_e)/4)$ are the excess charge and salt densities in the EDL [16, 20]. Eliminating i and j in (5.8) - (5.11) gives the governing equations for the SC model:

$$\epsilon\frac{\partial q}{\partial t} = \frac{\partial}{\partial x}\left((c+\epsilon w)\frac{\partial \psi}{\partial x}\right) + \epsilon\frac{\partial^2 q}{\partial x^2}, \tag{5.12}$$

$$\frac{\partial (c+\epsilon w)}{\partial t} = \frac{\partial^2 (c+\epsilon w)}{\partial x^2} + \epsilon\frac{\partial}{\partial x}\left(q\frac{\partial \psi}{\partial x}\right). \tag{5.13}$$

The total conductivity $(c + \epsilon w)$ includes contributions from the bulk (c) and the excess surface conductivity (ϵw), which becomes large at high potentials. The SC model quantitatively predicts the full DNS results (fig. 5.3b), and reveals that surface currents enhance the power density of porous electrodes.

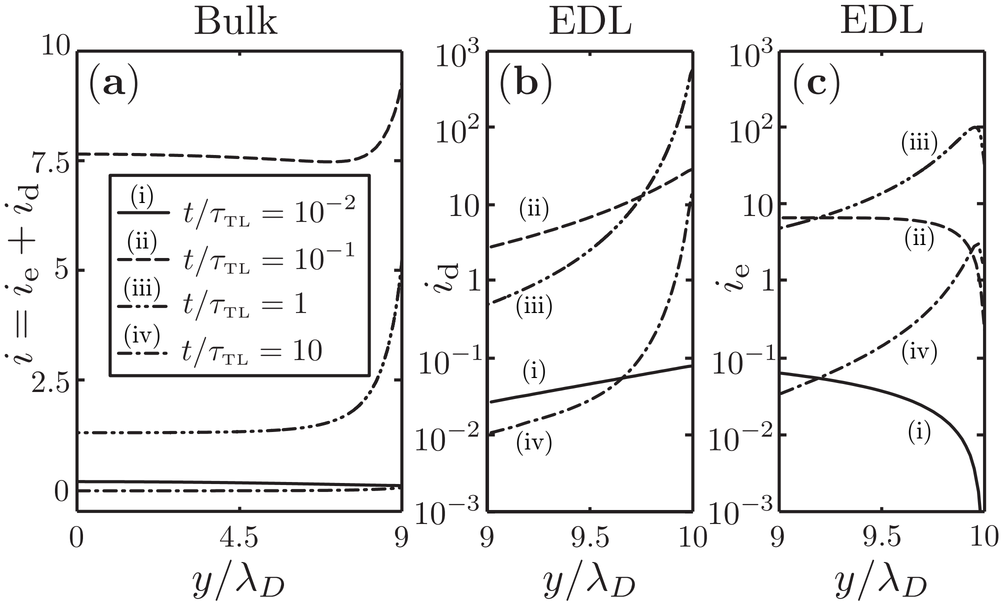


Figure 5.4: Axial current densities computed for $\Delta\psi = 7.5\,\psi_{\text{th}}$ in the middle of a single straight pore. (a) The total current in the bulk $(0 < y/\lambda_D < 9)$ rises and falls as the charging front passes. Within the EDL $(9 < y/\lambda_D < 10)$, the (b) diffusive and (c) Ohmic current densities also rise and fall, but are 2-3 orders of magnitude stronger, revealing the importance of surface conduction at high potentials.

5.4 Surface conduction as a short-circuit mechanism

To effectively enhance charging kinetics, surface conduction requires continuous conducting pathways in the charging direction to short circuit the high-resistance bulk. Whether such paths exist depends on the electrode morphology: surface conduction is much less effective in accelerating the charging of electrodes whose surfaces have discontinuities in the charging direction. Fig. 5.5 shows computed charging dynamics for a 'patchy' electrode, with chargeable segments held at a fixed potential (fig. 5.5a) separated by uncharged segments. These uncharged segments break up the continuous conducting pathways along the charging direction, and render surface conduction ineffective in short-circuiting the high-resistance bulk. In this case, the BB model captures the charging dynamics quantitatively. Straight (continuous) pores are accurately described by the SC model for multiple potentials, whereas patchy electrodes obey the slower BB charging kinetics (fig. 5.5d).

We conclude this chapter by noting that our Direct Numerical Simulation (DNS) of the Poisson-Nernst-Planck (PNP) equations have shown the classic Transmission Line (TL) model [51] to be effective in predicting the charging dynamics of electrodes with various morphologies, but only for low potentials. At higher potentials, charging dynamics are slowed dramatically due to the nonlinear capacitance of the EDL, as well as salt depletion of the pores. The BB model [20], which accounts for these nonlinear effects, *overpredicts* the charging time of electrodes with continuous conducting pathways along the charging direction. We have identified surface conductivity within the EDL as the mechanism responsible for 'short-circuiting' the high-resistance bulk, and shown that the BB approach, modified to include surface conduction, captures the charging dynamics quantitatively. Surface conduction can not accelerate the charging dynamics of electrodes whose surface morphologies have breaks in the charging direction (*e.g.* the patchy electrode of fig. 5.5), in which case the BB model captures the dynamics.

While electrode morphology has generally been regarded as irrelevant for charging dynamics, our results suggest an unanticipated boost in charging rates for electrodes with appropriate morphologies. One might expect conventional electrodes derived from activated carbon to contain sections that are 'dead-ends' from a surface conduction standpoint, in which case the slowed (BB) kinetics should hold. Electrodes derived from carbon nanotube forests [31], graphene sheets [161], or hierarchically-designed mesoporous carbons [137], on the other hand, naturally introduce the continuous surfaces required for SC 'short-circuits,' and may thus boost power densities.

Finally, let us finish by reflecting on the applicability of the PNP equations used in this chapter. The TL, BB, and SC models all assume the EDL to be thin, which may be violated in micropores (radius $< 2\,\mathrm{nm}$). Modified BB models [21, 22, 156, 120] have been developed based on the modified Donnan (mD) model [23] to describe charging kinetics in the micropores. Strictly speaking, the PNP equations are valid for dilute electrolytes in the mean-field limit, assumptions that are almost certainly violated at the high potentials of most applications or for ionic liquid electrolytes [56]. At such high potentials, the PNP equations predict unphysically high ion concentrations in the EDL [72], remedies for which have been proposed via steric repulsions between ions [72, 14, 61]. Given that the magnitude of the excess surface current varies with the ion concentration, any reduction in the EDL capacitance (*e.g.* due to steric repulsions [14], or Stern layers [21, 22]) will reduce the effect of surface conductivity. Such reductions affect the SC model as well as the BB model. Irrespective of the quantitative magnitude of the surface conductivity, however, the mechanism we have presented here – wherein excess surface conductivity short-circuits the electrode charging dynamics, boosting power densitiy – remains robust, even when steric or other effects reduce the magnitude.

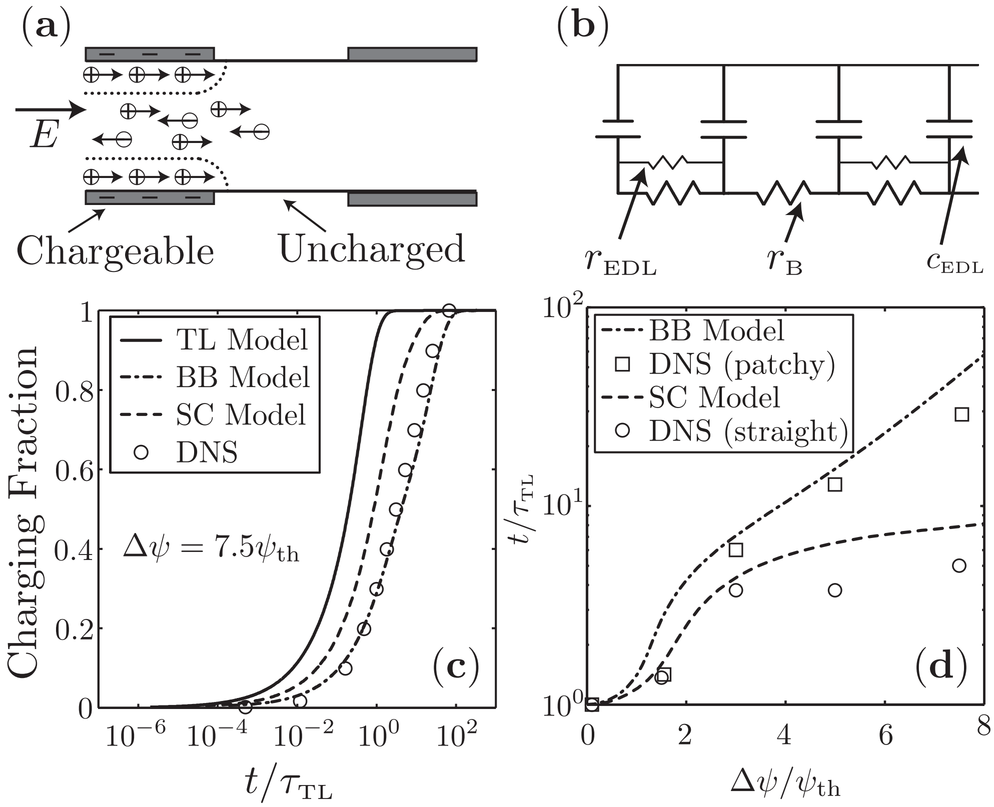


Figure 5.5: Charging kinetics of a patchy pore at $\Delta\psi = 7.5\psi_{\mathrm{th}}$. (a) A schematic of the patchy pore model in which only selected parts of the electrode surface may charge. (b) The patchy electrode's effective circuit introduces gaps in the low-resistance pathways introduced by surface conductivity, removing the 'short-circuit' that arose with straight pores (fig. 5.3d). (c) The lack of a continuous conducting paths in the charging direction renders surface conduction unable to short-circuit the high-resistance bulk. Omitting surface conduction recovers the original BB model, which captures the charging kinetics quantitatively. (d) Charging kinetics are slowed at high applied potentials, accurately described for straight pores by the SC model and for patchy pores by the BB model. The charging time is defined as that required for 99% charging ($\eta(t) = 0.99$).

Bibliography

[1] M. J. Aftosmis, M. J. Berger, and J. E. Melton. Adaptive Cartesian Mesh Generation. In *CRC Handbook of Mesh Generation (Contributed Chapter)*, 1998.

[2] S. A. Allison, J. J. Sines, and A. Wierzbicki. *J. Phys. Chem.*, 93:5819, 1989.

[3] A. S. Almgren, J. B. Bell, and P. Colella. *J. Comput. Phys.*, 46:1–46, 1998.

[4] U. M. Ascher and L. R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, 1998.

[5] U. M. Ascher, S. J. Ruuth, and B. T. R. Wetton. *SIAM J. Numer. Anal.*, 32:797, 1995.

[6] T. D. Aslam. *J. Comput. Phys.*, 193:349, 2004.

[7] N. Baker, M. Holst, and F. Wang. *J. Comp. Chem.*, 21:1343, 2000.

[8] N. A. Baker. *Curr. Opin. Struct. Biol.*, 15:137–43, 2005.

[9] N. A. Baker, D. Bashford, and D. A. Case. *New algorithms for macromolecular simulation.* Springer, 2006.

[10] N. A. Baker, D. Sept, S. Joseph, M. Holst, and J. McCammon. *P. Natl. Acad. Sci. U.S.A.*, 98:10037–41, 2001.

[11] A. J. Bard and L. R. Faulkner. *Electrochemical Methods: Fundamentals and Applications*. Wiley-Interscience, 2 edition, 2001.

[12] G. Bauer and V. Gravemeier. *Int. J. Numer. Meth. Eng.*, 86:1339–1359, 2011.

[13] M. Z. Bazant, K. T. Chu, and B. J. Bayly. *SIAM J. Appl. Math.*, 65:1463, 2005.

[14] M. Z. Bazant, M. Sabri Kilic, B. D. Storey, and A. Ajdari. *New J. Phys.*, 11:075016, 2009.

[15] M. Z. Bazant, B. Storey, and A. Kornyshev. *Phys. Rev. Lett.*, 106:046102, 2011.

[16] M. Z. Bazant, K. Thornton, and A. Ajdari. *Phys. Rev. E*, 70:1–24, 2004.

[17] H. E. Becker. U.S. patent 2,800,616 (to General Electric Co.), 1957.

[18] M. J. Berger and P. Colella. *J. Comput. Phys.*, 84:64–84, 1989.

[19] M. J. Berger and J. Oliger. *J. Comput. Phys.*, 53:484–512, 1984.

[20] P. M. Biesheuvel and M. Z. Bazant. *Phys. Rev. E*, 81:031502, 2010.

[21] P. M. Biesheuvel, Y. Fu, and M. Z. Bazant. *Phys. Rev. E*, 83:061507, 2011.

[22] P. M. Biesheuvel, Y. Fu, and M. Z. Bazant. *Russ. J. Electrochem.*, 48:580–592, 2012.

[23] P. M. Biesheuvel, S. Porada, M. Levi, and M. Z. Bazant. *J. Solid State Electrochem.*, 18:1365–1376, 2014.

[24] D. I. Boos. U.S. patent 3,536,963 (to Standard Oil, SOHIO), 1970.

[25] A. H. Boschitsch and M. O. Fenley. *J. Chem. Theory Comput.*, 7:1524–1540, 2011.

[26] D. Brogioli. *Phys. Rev. Lett.*, 103:31–34, 2009.

[27] T. Brumleve and R. Buck. *J. Electroanal. Chem.*, 90:1–31, 1978.

[28] H. Bruus. *Theoretical Microfluidics.* Oxford University Press, 2008.

[29] M. Buoni and L. Petzold. *J. Comput. Phys.*, 225:2320–2332, 2007.

[30] M. Buoni and L. Petzold. *J. Comput. Phys.*, 229:379–398, 2010.

[31] Z. G. Cambaz, G. Yushin, S. Osswald, V. Mochalin, and Y. Gogotsi. *Carbon*, 46:841–849, 2008.

[32] T. Can, C.-I. Chen, and Y.-F. Wang. *J. Mol. Graph. Model*, 25:442–454, 2006.

[33] A. E. Cárdenas, R. D. Coalson, and M. G. Kurnikova. *Biophys. J.*, 79:80–93, 2000.

[34] D. Chen, J. Lear, and B. Eisenberg. *Biophys. J.*, 72:97–116, 1997.

[35] H. Chen, C. Min, and F. Gibou. *J. Sci. Comput.*, 31:19, 2007.

[36] J. Chen, C. L. Brooks, and J. Khandogin. *Curr. Opin. Struct. Biol.*, 18:140–8, 2008.

[37] L. Chen, M. J. Holst, and J. Xu. *SIAM J. Numer. Anal.*, 45(6):2298–2320, 2007.

[38] I.-L. Chern, J.-G. Liu, and W. Wang. *Methods and Applications of analysis*, 10:309–328, 2003.

[39] D. Choi, G. Blomgren, and P. Kumta. *Adv. Mater.*, 18:1178–1182, 2006.

[40] T. Christen. *J. Power Sources*, 91:210–216, 2000.

[41] H. Cohen and J. W. Cooley. *Biophys. J.*, 5:145–162, 1965.

[42] M. L. Connolly. *J. Appl. Crystallogr.*, 16:548–558, 1983.

[43] M. L. Connolly. *J. Mol. Graphics*, 11:139–141, 1993.

[44] B. E. Conway. *J. Electrochem. Soc.*, 138, 1991.

[45] B. E. Conway. *Electrochemical supercapacitors: scientific fundamentals and technological applications.* Kluwer Academic/Plenum Publishers, 1999.

[46] B. E. Conway, V. Birss, and J. Wojtowicz. *J. Power Sources*, 66:1, 1997.

[47] C. M. Cortis and R. A. Friesner. *J. Comp. Chem.*, 18:1570, 1997.

[48] C. M. Cortis and R. A. Friesner. *J. Comp. Chem.*, 18:1591, 1997.

[49] E. L. Cussler. *Diffusion: mass transfer in fluid systems.* Cambridge, 3 edition, 2009.

[50] M. E. Davis and J. A. McCammon. *J. Comp. Chem.*, 10:386, 1989.

[51] R. de Levie. *Electrochim. Acta*, 8:751–780, 1963.

[52] E. A. Demekhin, V. S. Shelistov, and S. V. Polyanskikh. *Phys. Rev. E*, 84:036318, 2011.

[53] E. Dowgiallo and J. Hardin. *IEEE Aero. El. Sys. Mag.*, 10:26–31, 1995.

[54] M. Eikerling, A. A. Kornyshev, and E. Lust. *J. Electrochem. Soc.*, 152:E24, 2005.

[55] R. S. Eisenberg. *J. Membr. Biol.*, 150:1–25, 1996.

[56] K. F., T. Tamm, M. M. Hantel, E. Perre, A. Aabloo, E. Lust, M. Z. Bazant, and V. Presser. *Electrochem. Commun.*, 34:196–199, 2013.

[57] F. Fogolari, A. Brigo, and H. Molinari. *J. Mol. Recognit.*, 15 (6):377–392, 2002.

[58] E. Frackowiak and F. Beguin. *Carbon*, 39:937, 2001.

[59] W. Geng, S. Yu, and G. Wei. *J. Chem. Phys*, 127(11):114106, 2007.

[60] F. Gibou, C. Min, and R. Fedkiw. *J. Sci. Comput.*, 54:369–413, 2012.

[61] B. Giera, N. Henson, E. M. Kober, T. M. Squires, and M. S. Shell. *Phys. Rev. E*, 88:011301(R), 2013.

[62] M. K. Gilson, K. a. Sharp, and B. H. Honig. *J. Comput. Chem.*, 9:327–335, 1988.

[63] M. K. Gilson, K. A. Sharp, and B. H. Honig. *J. Comp. Chem.*, 9:327, 1988.

[64] J. Greer and B. L. Bush. *P. Natl. Acad. Sci. U.S.A.*, 75:303, 1978.

[65] A. Helgadottir and F. Gibou. *J. Comput. Phys.*, 230:3830–3848, 2011.

[66] M. Holst, N. Baker, and F. Wang. *J. Comp. Chem.*, 21:1319, 2000.

[67] M. Holst and F. Saied. *J. Comp. Chem.*, 14:105, 1993.

[68] M. J. Holst and F. Saied. *J. Comp. Chem.*, 16:337, 1994.

[69] M. Itagaki, S. Suzuki, I. Shitanda, K. Watanabe, and H. Nakazawa. *J. Power Sources*, 164:415–424, 2007.

[70] B. Jayaram, K. A. Sharp, and B. Honig. *Biopolymers*, 28:975, 1989.

[71] A. H. Juffer, E. F. F. Botta, B. A. M. van Keulen, A. van der Ploeg, and H. J. C. Berendsen. *J. Comput. Phys.*, 97:144, 1991.

[72] M. Kilic, M. Z. Bazant, and A. Ajdari. *Phys. Rev. E*, 75:1–16, 2007.

[73] J. Kirkwood. *J. Chem. Phys.*, 2:351, 1934.

[74] P. Koehl. *Curr. Opin. Struct. Biol.*, pages 142–151, 2006.

[75] R. Kötz. *Electrochim. Acta*, 45:2483–2498, 2000.

[76] R. Kötz and M. Carlen. *Electrochim. Acta*, 45:2483, 2000.

[77] M. G. Kurnikova, R. D. Coalson, P. Graf, and a. Nitzan. *Biophys. J.*, 76:642–56, 1999.

[78] Y. K. Kwok and C. C. K. Wu. *Numer. Meth. Part. D. E.*, 11:389–397, 1995.

[79] B. Lee and F. Richards. *J. Mol. Biol.*, 55:379–400, 1971.

[80] H. Lee. *J. Solid State Chem.*, 144:220–223, 1999.

[81] S. M. Lee, D. C. Chung, D. J. Bae, S. C. Lim, and Y. H. Lee. *Adv. Mater.*, 13:479, 2001.

[82] D. G. Levitt. *Biophys. J.*, 59:271–277, 1991.

[83] J. D. J. A. M. Li-Tien Cheng and B. Li. *J. Chem. Phys.*, 127:084503, 2007.

[84] J. Lim, J. D. Whitcomb, J. G. Boyd, and J. Varghese. *Comput. Mech.*, 43:461–475, 2008.

[85] H. Lin, B. D. Storey, M. H. Oddy, C.-H. Chen, and J. G. Santiago. *Phys. Fluids*, 16:1922, 2004.

[86] K. Liu and M. A. Anderson. *J. Electrochem. Soc.*, 143:124, 1996.

[87] F. Losasso, R. Fedkiw, and S. Osher. *Comput. Fluids*, 35:995–1010, 2006.

[88] F. Losasso, F. Gibou, and R. Fedkiw. *ACM T. Graphic.*, 23:457–462, 2004.

[89] B. Lu, M. J. Holst, J. Andrew McCammon, and Y. C. Zhou. *J. Comput. Phys.*, 229:6979–6994, 2010.

[90] B. Lu, Y. Zhou, M. Holst, and J. McCammon. *Commun. Comput. Phys.*, 3 (5):973–1009, 2008.

[91] R. Luo, L. David, and M. K. Gilson. *J. Comp. Chem.*, 23:1244, 2002.

[92] B. A. Luty, M. E. Davis, and J. A. McCammon. *J. Comp. Chem.*, 13:1114, 1992.

[93] J. Lyklema. *Fundamentals of Interface and Colloid Science*, volume 2. Academic Press, 1995.

[94] K. Machida, K. Furuuchi, M. Min, and K. Naoi. *Electrochemistry*, 72:402, 2004.

[95] J. A. Manzanares and W. D. Murphy. *J. Phys. Chem.*, 97:8524–8530, 1993.

[96] S. McLaughlin. *Annu. Rev. Biophys. Biophys. Chem.*, 18:113, 1989.

[97] A. M. Micu, B. Bagheri, A. V. Ilin, L. R. Scott, and B. M. Pettitt. *J. Comput. Phys.*, 136:263–271, 1997.

[98] J. Miller and P. Simon. *Science*, 321:651, 2008.

[99] C. Min and F. Gibou. *J. Comput. Phys.*, 225:300–321, 2007.

[100] C. Min and F. Gibou. *J. Comput. Phys.*, 226:1432–1443, 2007.

[101] C. Min, F. Gibou, and H. D. Ceniceros. *J. Comput. Phys.*, 218:123, 2006.

[102] M. Mirzadeh, M. Theillard, and F. Gibou. *J. Comput. Phys.*, 230:2125–2140, 2011.

[103] M. Mirzadeh, M. Theillard, A. Helgadottir, D. Boy, and F. Gibou. *Commun. Comput. Phys.*, 13:150–173, 2013.

[104] W. D. Murphy and J. A. Manzanares. *J. Phys. Chem.*, 9991:9983–9991, 1992.

[105] K. Naoi, S. Suematsu, and A. Manago. *J. Electrochem. Soc.*, 147:420, 2000.

[106] J. Newman and K. E. Thomas-Alyea. *Electrochemical Systems*. Wiley-Interscience, 3 edition, 2004.

[107] J. Newman and W. Tiedemann. *AIChE J.*, 21:25–41, 1975.

[108] Y.-T. Ng, C. Min, and F. Gibou. *J. Comput. Phys.*, 228:8807–8829, 2009.

[109] A. Nicholls and B. Honig. *J. Comp. Chem.*, 12:435, 1991.

[110] R. Nilson and S. Griffiths. *Phys. Rev. E*, 80:1–10, 2009.

[111] C. Niu, E. K. Sichel, R. Hoch, D. Moy, and H. Tennent. *Appl. Phys. Lett.*, 70:1480, 1997.

[112] R. W. O'Brien and L. R. White. *J. Chem. Soc. Farad. T 2*, 74:1607, 1978.

[113] R. W. O'Brien and L. R. White. *J. Chem. Soc., Faraday Trans. 2*, 74:1607–1626, 1978.

[114] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces.* Springer, 2003.

[115] S. Osher and J. Sethian. *J. Comput. Phys.*, 79:12–49, 1988.

[116] G. W. P. Bates and S. Zhao. *J. Comput. Chem.*, 29:380–391, 2008.

[117] Q. Pan and X.-C. Tai. *Commun. Comput. Phys.*, 6:777–792, 2009.

[118] J. Papac, F. Gibou, and C. Ratsch. *J. Comput. Phys.*, 229:875–889, 2010.

[119] R. Pekala. *J. Non-Cryst. Solids*, 225:74–80, 1998.

[120] S. Porada, L. Borchardt, M. Oschatz, M. Bryjak, J. S. Atchison, K. J. Keesman, S. Kaskel, P. M. Biesheuvel, and V. Presser. *Energy Environ. Sci.*, 6:3700, 2013.

[121] S. Porada, R. Zhao, A. van der Wal, V. Presser, and P. M. Biesheuvel. *Prog. Mater. Sci.*, 58:1388–1442, 2013.

[122] D. Qu and H. Shi. *J. Power Sources*, 74:99–107, 1998.

[123] R. O'Hayre, S. Cha, W. Colella, and F. B. Prinz. *Fuel Cell Fundamentals.* John Wiley & Sons, 2005.

[124] A. A. Rashin and J. Malinsky. *J. Comp. Chem.*, 12:981, 1991.

[125] R. Rica, R. A. Ziano, D. Salerno, F. Mantegazza, and D. Brogioli. *Phys. Rev. Lett.*, 109:156103, 2012.

[126] R. A. Rica, D. Brogioli, and R. Ziano. *J. Phys. Chem. C*, 116:16934–16938, 2012.

[127] F. Richards. *Annu. Rev. Biophys. Bio.*, 6:151–176, 1977.

[128] A. Rudge, J. Davey, I. Raistrick, S. Gottesfeld, and J. Ferraris. *J. Power Sources*, 47:89–107, 1994.

[129] W. B. Russel, D. A. Saville, and W. R. Schowalter. *Colloidal Dispersions.* Cambridge University Press, 1989.

[130] H. Sakaguchi and R. Baba. *Phys. Rev. E*, 76:011501, 2007.

[131] M. F. Sanner and A. J. Olson. *Biopolymers*, 38:305–320, 1996.

[132] S. Sarangapani. *J. Electrochem. Soc.*, 143:3791, 1996.

[133] J. A. Sethian. *Level Set Methods and Fast Marching Methods.* Cambridge University Press, 2000.

[134] K. A. Sharp and B. Honig. *Annu. Rev. Biophys. Biophys. Chem.*, 19:301, 1990.

[135] K. A. Sharp and B. Honig. *J. Phys. Chem.*, 94:7684–7692, 1990.

[136] P. Simon and Y. Gogotsi. *Nat. Mater.*, 7:845–54, 2008.

[137] P. Simon and Y. Gogotsi. *Nat. Mater.*, 7:845–854, 2008.

[138] T. M. Squires. *Rev. Mod. Phys.*, 77:977–1026, 2005.

[139] T. M. Squires and M. Z. Bazant. *J. Fluid Mech.*, 509:217–252, 2004.

[140] V. Srinivasan and J. W. Weidner. *J. Electrochem. Soc.*, 144:L210, 1997.

[141] M. D. Stoller, S. Park, Y. Zhu, J. An, and R. S. Ruoff. *Nano Lett.*, 8:3498–502, 2008.

[142] J. Strain. *J. Comput. Phys.*, 151:616, 1999.

[143] M. E. Suss, T. F. Baumann, W. L. Bourcier, C. M. Spadaccini, K. a. Rose, J. G. Santiago, and M. Stadermann. *Energ. Environ. Sci.*, 5:9511, 2012.

[144] M. E. Suss, P. M. Biesheuvel, T. F. Baumann, M. Stadermann, and J. G. Santiago. *Environ. Sci. Technol.*, 48:2008 – 2015, 2014.

[145] R. S. T.C. Murphy, R.B. Wright. In F.M. Del-nick, D. Ingersoll, X. Andrieu, K. Naoi, editor, *Electro-chemical Capacitors II*, volume 96-25, page 258. The Electrochemical Society, 1997.

[146] M. Toupin, T. Brousse, and D. Bélanger. *Chem. Mater.*, 16:3184–3190, 2004.

[147] J. Wang, L. Angnes, H. Tobias, R. A. Roesner, K. C. Hong, R. S. Glass, F. M. Kong, and R. W. Pekala. *Anal. Chem.*, 65:2300–2303, 1993.

[148] J. Warwicker and H. C. Watson. *J. Mol. Biol.*, 157:671–679, 1982.

[149] G.-W. Wei, Q. Zheng, Z. Chen, and K. Xia. *SIAM Rev.*, 54:699–754, 2012.

[150] R.-J. Yang, L.-M. Fu, and Y.-C. Lin. *J. Colloid Interface Sci.*, 239:98, 2001.

[151] B. J. Yoon and A. M. Lenhoff. *J. Comp. Chem.*, 11:1080, 1990.

[152] S. Yu and G.-W. Wei. *J. Comput. Phys.*, 227:602–632, 2007.

[153] S. Yu, Y. Zhou, and G.-W. Wei. *J. Comput. Phys.*, 224:729–756, 2007.

[154] G. Yushin, Y. Gogotsi, C. Portet, P. Simon, and P. L. Taberna. *Science*, 313:1760, 2006.

[155] T. A. Zangle, A. Mani, and J. G. Santiago. *Langmuir*, 25:3909–16, 2009.

[156] R. Zhao, M. van Soestbergen, H. H. M. Rijnaarts, a. van der Wal, M. Z. Bazant, and P. M. Biesheuvel. *J. Colloid Interface Sci.*, 384:38–44, 2012.

[157] J. P. Zheng. *J. Electrochem. Soc.*, 142:2699, 1995.

[158] J. P. Zheng and T. R. Jow. *J. Power Sources*, 62:155, 1996.

[159] Q. Zheng, D. Chen, and G.-W. Wei. *J. Comput. Phys.*, 230:5239–5262, 2011.

[160] Y. C. Zhou, M. Feig, and G.-W. Wei. *J. Comput. Chem.*, 29:87–97, 2007.

[161] Y. Zhu, S. Murali, M. D. Stoller, K. J. Ganesh, W. Cai, P. J. Ferreira, A. Pirkle, R. M. Wallace, K. A. Cychosz, M. Thommes, D. Su, E. A. Stach, and R. S. Ruoff. *Science*, 332:1537–1541, 2011.