Non-linear Electrophoresis of Ideally Polarizable Particles

by

Wai Hong Ronald Chan

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Bachelor of Science in Engineering as Recommended by the Department of Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2014

Author .....
Department of Mechanical Engineering
May 9, 2014

Certified by .....
Cullen R. Buie
Assistant Professor
Thesis Supervisor

Accepted by .....
Anette Hosoi
Professor of Mechanical Engineering
Undergraduate Officer

Non-linear Electrophoresis of Ideally Polarizable Particles

by

Wai Hong Ronald Chan

Submitted to the Department of Mechanical Engineering
on May 9, 2014, in partial fulfillment of the
requirements for the degree of
Bachelor of Science in Engineering as Recommended by the Department of
Mechanical Engineering

Abstract

This thesis investigates the non-linear regime of electrophoresis, in particular the variation of electrophoretic velocity with electric field at high field strengths. Known theoretical approaches to the problem accounting for ion steric effects, dielectric decrement effects and charge-induced thickening are consolidated, further developed and validated using numerical simulations. In doing so, the influences of the relative strengths of surface conductivity and bulk conductivity and of the relative importance of advection to diffusive transport in the electrolyte are both investigated. In addition, further light is shed on the dependence of electrophoretic mobility on the ionic and particle sizes, and on the relevant ionic diffusivities.

Thesis Supervisor: Cullen R. Buie
Title: Assistant Professor

Acknowledgments

I would like to express my sincere gratitude to my thesis supervisor, Professor Cullen Buie, for his support and advice over the past two years. In addition, I would like to thank Professor Bruno Figliuzzi and Dr Jeffrey Moran for their guidance and insights. Without their supervision and constant help, this thesis would not have been possible.

I would like to thank the Agency for Science, Technology and Research, Singapore, for their financial support throughout my undergraduate education. I would also like to thank the Undergraduate Research Opportunities Program (UROP) office and the group members of the Laboratory for Energy and Microsystems Innovation at the Massachusetts Institute of Technology for their continued support of this project.

In addition, I would like to thank my parents for their encouragement all these years, and the faculty of the Department of Mechanical Engineering for their valuable guidance throughout my undergraduate years. Last but not least, I am grateful to my friends for their support through thick and thin, including Edwin Khoo, Kaicheng Liang, Sarandeth Reth, Yang Chen, David Kang, Jodie Sun, Yun Liu, Kun Xue, and many others.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

Electrophoresis is defined as the motion of particles suspended in a fluid under the influence of an externally applied electric field. In this thesis, we focus on the non-linear electrophoresis of ideally polarizable charged particles. A schematic illustrating such a particle is shown in Figure 1-1. At large applied field strengths, significant ionic exchange occurs between the electric double layer, which is essentially a condensed layer of counterions surrounding each charged particle, and the bulk solution. In addition, steric effects due to the finite size of ions have an impact on the electric potential distribution in the electric double layer. In this case, the velocity field, the electric potential and the ionic concentration around the particle are described by a complicated set of coupled non-linear partial differential equations, and have to be solved numerically. These equations could be modified to account for the concentration dependence of the dielectric permittivity and the dynamic viscosity of the electrolyte, as well as the asymmetric diffusivities and sizes of the coions and counterions in the electrolyte, thereby obtaining a more realistic physical model of the situation.

To solve these equations, a pseudo-spectral method was employed. Such a method was used successfully by Chu and Bazant [2, 3] to determine the electric potential and the ionic concentration around an ideally polarizable metallic sphere. The numerical simulations that were performed incorporate the steric model developed by Kilic *et al.* [4, 5] to account for crowding effects in the electric double layer, advective

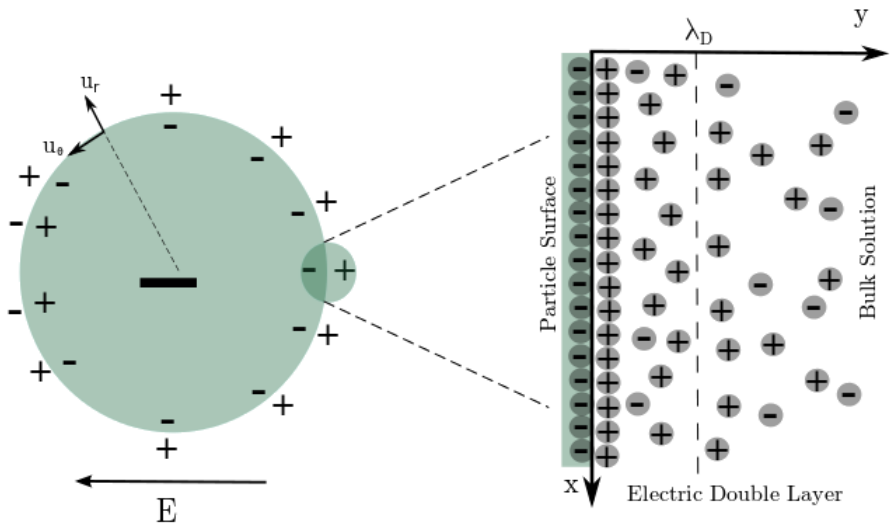

Figure 1-1: Schematic representation of the electrophoresis of an ideally polarizable particle of negative charge. In the thin double layer approximation, the calculation is performed by considering two asymptotic solutions in the bulk solution (left) and in the electric double layer (right).

transport, and the presence of a body force in the bulk electrolyte. In addition, they incorporate the dielectric decrement model first used in this context by Hatlo *et al.* [1] to account for the decrease in dielectric constant of a solvent in regions with high ionic concentrations due to the polarization of the solvent molecules caused by the ions in their vicinity.

1.1 Electrophoresis

The mathematical description of electrophoresis of particles with thin electric double layers builds upon the classical theory developed by Smoluchowski [7] in 1903, which considers dielectric particles with immobile surface charges. Smoluchowski noted that the net force exerted by an externally applied electric field on a system consisting of a suspended spherical particle and the counterion cloud surrounding it should be zero since the system does not have a net total charge. However, the co-existence of a charged region in the electric double layer and of an externally applied field leads to the appearance of an electro-osmotic flow, which is responsible for the motion of the particle in the electrolyte. Smoluchowski demonstrated that the velocity $\mathbf{u}_{EP}$ of the

charged particle in the frame of the electrolyte is proportional to the applied field E

$$\mathbf{u}_{EP} = \mu_{EP}\mathbf{E}, \tag{1.1}$$

where μ_{EP} denotes the electrophoretic mobility of the particle. Smoluchowski's formula is only valid for particles with low zeta potentials. As demonstrated by Morrison [8] and Anderson [9], the formula can also be applied to non-spherical particles. However, the formula breaks down at higher voltages due to non-linear effects arising in the double layer. In this scenario, the electrophoretic velocity of the particle would no longer be linearly proportional to the applied electric field.

Electrophoresis provides an efficient way to manipulate charged particles in microfluidic devices. Applications include lab-on-a-chip technologies [10, 11] and DNA transport and separation [12].

1.2 High field strengths: Non-linear regime

As large applied electric fields are commonly encountered in these microfluidic devices, the determination of the mobility μ_{EP} at high fields has been a key focus in colloid science for several decades [13, 14].

Bikerman [15] showed in 1940 that the surface conductivity of the electric double layer cannot be neglected in comparison with the conductivity of the bulk solution at large zeta potentials. The non-linear regime is then characterized by a net exchange of ions between the electric double layer and the bulk solution. This results in local ion depletion and accumulation in the electrolyte in the immediate vicinity of the particle.

Overbeek [16], Dukhin and co-workers [17, 18, 19, 20, 21, 22, 23, 24], and O'Brien and collaborators [25, 26, 27] have made significant progress towards the understanding of non-linear electrophoresis. Dukhin and his collaborators [17, 18, 19, 20, 21, 22, 23, 24] have notably recognized that the electrophoretic mobility generally depends on the applied electric field, and have described the key role played by concentra-

tion polarization in electrokinetic phenomena at high applied field strengths. In particular, they have demonstrated that concentration polarization results in a fluid motion driven by concentration gradients. This phenomenon, referred to as diffusio-osmosis, was independently rediscovered by Prieve in 1984 [28]. The combination of electro-osmosis and diffusio-osmosis leading to particle motion has been referred to as diffusio-phoresis and was subsequently investigated by Zaltzman and Rubinstein [29, 30], and by Rica and Bazant [31]. In their 2010 study, Rica and Bazant considered the situation where an electrolyte is subject to a direct current electric field. In this case, concentration gradients arise in the bulk solution, resulting in both electro- and diffusio-osmotic contributions to the flow in the immediate vicinity of colloidal particles suspended in the electrolyte. Their analysis relies on the thin double layer approximation and neglects both surface conduction and advective transport in the bulk solution.

Most of the studies on the non-linear electrophoresis of particles with fixed surface charge rely on weak field approximations, which enable a linearization of the model equations. Recently, Schnitzer and Yariv [32, 33, 34] have developed analytical models aimed at describing the strongly non-linear regime for particles with thin electric double layers. At asymptotically large applied voltages, they demonstrated that advective transport in the bulk electrolyte results in a uniform salt concentration. They also showed that the asymptotic matching between the current density emerging from the electric double layer and the current density in the electro-neutral bulk solution is incompatible with the asymptotic matching between the corresponding salt fluxes. To resolve this apparent contradiction, they proposed the emergence of a diffuse boundary layer in the overlap region of the electric double layer and the bulk electrolyte.

1.3 Ideally polarizable particles

Non-linear electrokinetics experienced a renewed interest in the mid-1990s, when various non-linear electrokinetic phenomena were discovered and explored, including

alternating current electro-osmosis [35] and induced-charge electrokinetic phenomena around conducting colloidal particles [36, 37, 38]. Ideally polarizable particles are one of the key components of induced-charge electrokinetics. These particles are ideal conductors, thus enabling a recombination of their surface charges when an electric field is applied. As a result, the zeta potential varies on the surface of the particle and an induced electric field appears in the bulk solution.

Electro-osmotic flows around a perfect conductor have been widely studied, notably by Squires and Bazant [37, 38], who suggested several applications in microfluidic devices, and Chu and Bazant [3], who conducted numerical simulations to determine the electric potential and ionic concentration around a conducting sphere, as mentioned earlier. Bazant *et al.* [6] and Yariv [39] have subsequently studied induced-charge electrophoresis. Relying on the Gouy-Chapman model to describe the electric double layer, Yariv obtained an asymptotical expression relating the electrophoretic mobility to the applied electric field, which states that the electrophoretic mobility tends to zero at large applied electric fields. Such a trend has not been observed experimentally. The decay of the mobility with the applied electric field in Yariv's formula can be attributed to the inability of the Gouy-Chapman model to adequately describe the physics of the electric double layer at large applied field strengths [4, 5, 40]. More recently, Schnitzer and Yariv have derived analytical macroscale models accounting for surface conduction mechanisms in induced-charge electro-osmosis [41], in one of the first attempts to incorporate surface conduction in an analytic description of induced-charge electrokinetic phenomena.

1.4 Asymptotic matching and the steric model

In studying electrophoresis, the physical domain is classically divided into two regions: the bulk and the electric double layer. Both regions are described by asymptotic solutions, which are matched using appropriate boundary conditions.

Several models have been developed to describe the electric double layer. The Gouy-Chapman model, developed independently in 1910 by Gouy [42] and in 1913

by Chapman [43], is usually the starting point of all attempts to describe non-linear effects in the electric double layer. This model relies on dilute solution theory, where ions are described as non-interacting point charges. However, at high zeta potentials, the Gouy-Chapman model predicts diverging ionic concentrations in the electric double layer close to the particle surface.

A first model aimed at correcting the non-physical divergence of the Gouy-Chapman description was proposed by Stern [44] in 1924. Stern introduced the concept of a compact layer. As reviewed by Bazant *et al.* [6], the first model accounting for steric effects due to the finite size of ions in the electric double layer was developed by Bikerman [40] in 1942. Bikerman's model was rediscovered in 2007 by Kilic *et al.* [4, 5], who investigated the influence of steric effects on the charge dynamics and conductivity of the electric double layer. Bazant *et al.* later identified the importance of steric effects in induced-charge electrophoresis. Accounting for ionic steric effects, they found an asymptotic expression demonstrating that the electrophoretic mobility scales as the square root of the applied electric field at large voltages. This result fundamentally changes the prediction of Yariv [39], which relies on dilute solution theory to describe the electric double layer at high voltages. The model of Kilic *et al.* [4, 5] has been subsequently applied by Khair and Squires [45] to study ion steric effects on the electrophoresis of dielectric particles with uniform surface charge.

1.5 Dielectric decrement

The role of ion hydration, or the accumulation of solvent molecules around ions, in modifying the dielectric constant of an electrolyte and the structure of the electric double layer was first recognized by Bikerman [40] in 1942. A variation in dielectric permittivity and a change in the ionic concentration profile in the electric double layer would both result in changes to the electrophoretic mobility predicted by the model to be discussed in Chapter 2. Bikerman noted that a difference between the dipole moments of the ions and the solvent molecules would result in changes to the physical interactions between the ions and the solvent molecules. This would in turn

modify the response of the solvent to an externally applied electric field, as well as the distribution of ions in the double layer. To account for this, Bikerman proposed modifications of the Poisson-Boltzmann equation to account for ion-solvent interactions. However, he did not furnish many details due to the lack of computational power back then, and this idea was again not picked up until much later.

Meanwhile, there has been increased recognition over the years of ion-specific effects in various physical phenomena. These effects turn out to be reasonably well described by the well-known Hofmeister series [46], which characterizes the extent to which an ion affects the solubility of a protein in the same solution. Analytical and computational approaches remain less straightforward than phenomenological approaches in describing ion-specific effects in a more quantitative manner [47], and thus empirical correlations have been commonly used to characterize these effects. One such empirical correlation exists in characterizing ion-specific effects on dielectric permittivity: several authors [48, 49] have noted, based on the analysis of empirical data, that at ionic concentrations of under 2M, the following linear relation can be used to determine the dielectric constant of an ionic solution

$$\epsilon_m = [\epsilon_w - \alpha(c_+ + c_-)]\epsilon_0 \tag{1.2}$$

where ϵ_m is the dielectric permittivity of the solution, ϵ_w is the relative permittivity of the solvent (water), ϵ_0 is the permittivity of free space, $c_\pm$ are the cationic and anionic concentrations, and α is a phenomenological coefficient termed the excess ion polarization. These authors have compiled a list of values of α for different ions. (Attempts have also been made [50, 51, 52] to derive this coefficient for some ions from first principles and using computational techniques.) Since the presence of ions reduces the permittivity of the solution, this phenomenon has been termed the "dielectric decrement" effect.

Adding the dielectric decrement effect to an electrokinetics model would suggest that particles in different ionic electrolytes with different ionic concentrations would have different electrophoretic mobilities, since the dielectric permittivity of the elec-

trolyte has an impact on the electrophoretic mobility of particles suspended in it. This has indeed been observed experimentally by several authors [6, 53, 54, 55]. The ion-specific nature of electrophoresis and the additional dependence on ionic concentration lend weight to Bikerman's intuition that ion-solvent interactions could have an effect on electrophoretic mobility. (While Bazant *et al.* [6] identify a decreased dielectric permittivity in the double layer as a potential reason for ion specificity, the review focuses on the field dependence rather than the concentration dependence of the dielectric permittivity, and thus sees the decreased dielectric permittivity as having a less significant impact than that of steric effects.)

Based on the phenomenological relation for the dielectric permittivity above and a mean-field approximation, Ben-Yaakov *et al.* [47, 56] derived a modified Poisson-Boltzmann relation that accounts for the dielectric decrement effect. The same relation was later obtained in a more explicit form by Frydel [57]. Subsequently, Hatlo *et al.* [1] combined the dielectric decrement effect with the steric model of Kilic *et al.* [4, 5] to obtain an extended modified Poisson-Boltzmann relation. (López-García [58] was able to derive a modification of the Poisson-Boltzmann relation of a similar form by using polarizable insulating spheres to model ions. However, the model requires knowledge of the permittivity of the spheres. In addition, a mean field approximation is not adopted in this work, making it difficult to numerically evaluate the derived expression.) Most recently, Zhao and Zhai [59] extended the dielectric decrement model (without steric effects) to fixed-charge electrokinetics and were able to determine the electrophoretic mobility for suspended dielectric colloidal particles under the thin double layer approximation.

1.6 Charge-induced thickening

In their 2009 review, Bazant *et al.* [6] identified the possibility that the crowding of counterions in the electric double layer could increase the local viscosity. This would, in turn, result in a modification of the predicted electrophoretic mobility. Due to the substantial literature involved in this field, we will defer a full discussion

of the characterization of this viscosity increase to Chapter 4. At the moment, it suffices to remark that similar to the dielectric decrement case, a range of analytic, computational and empirical approaches have been taken to model the concentration dependence of dynamic viscosity to varying success rates; however, we propose that a general expression involving the local volume fraction of ions can be used to model charge-induced thickening to reasonable confidence without the need for a phenomenological coefficient.

1.7 Organization of thesis

In this thesis, we seek to determine the variation of electrophoretic velocity with electric field at high field strengths, taking into account such effects as steric repulsion, dielectric decrement, charge-induced thickening, surface conduction and advection.

Parts of the introduction and Chapters 2 and 6 of this thesis are based on a previous work co-authored by this author with Figliuzzi, Moran and Buie[1]. In addition, Chapter 4 of this work is inspired by the analysis of Bazant *et al.* [6], which calls for greater attention to the concentration dependence of the dynamic viscosity of the electrolyte.

In Chapter 2, the mathematical model used to find the solution of the electrokinetics equations in the electric double layer is described more comprehensively. For the characterization of the electric double layer, a modified Poisson-Boltzmann equation following the steric model of Kilic *et al.* [4, 5] is used to obtain a modified electric potential distribution. At the bulk-electric double layer interface, the electrophoretic slip velocity is expressed as a combination of electro-osmosis and diffusio-osmosis, and effective boundary conditions are used to match the asymptotic solutions in the electric double layer and in the bulk electrolyte. In the bulk electrolyte, the contributions of diffusion, electromigration and advection to the ionic fluxes are considered.

In Chapter 3, the steric model is extended to include dielectric decrement effects in

[1]Figliuzzi, M., Chan, W. H. R., Moran, J., Buie, C., Nonlinear electrophoresis of ideally polarizable particles, submitted to *Physics of Fluids*.

the electric double layer. In addition, two simplifications to the model are described. Firstly, a simplified model in which diffusio-osmosis is neglected is used to compare the relative importance of steric effects and dielectric decrement effects. Secondly, a condensed layer model for the electric double layer is proposed. In this composite model, an analytic asymptotic expression for the electrophoretic mobility at high field strengths can be derived.

In Chapter 4, the motivation for investigating the concentration dependence of the dynamic viscosity of the electrolyte is described in greater detail, and some expressions that could be used to model this concentration dependence are suggested. These expressions have been derived by other authors both from first principles and from empirical data.

In Chapter 5, the non-linear coupled equations described earlier in Chapter 2 are modified to account for the asymmetry in ionic diffusivities and sizes in the electrolyte. The implications of these asymmetries are then discussed.

In Chapter 6, the numerical methods used to solve the coupled non-linear equations are described in greater detail, and interesting trends and observations from the numerical simulations that were performed are discussed. In particular, we examine the weakly nonlinear regime, and present the results of the simulations in situations where surface conduction, advection, crowding effects, dielectric decrement effects and charge-induced thickening effects are significant.

Finally, we conclude with the key findings of this thesis in Chapter 7.

Chapter 2

Basic theoretical model including steric effects

2.1 Electric double layer

2.1.1 Electric potential

In this section, we rely on an earlier model developed by Kilic *et al.* [4, 5] to describe the electric potential distribution within the electric double layer (EDL). In this model, the total Helmholtz free energy $\mathcal{F}$ in the EDL can be expressed as the sum of the electrostatic energy $\mathcal{U}$ and of an entropic contribution $-T\mathcal{S}$. As a result, the electric potential is effectively determined by the local mean charge density. The electrostatic energy can be written as a function of the excess electric potential ϕ, defined as the electric potential difference $\psi - \psi_b$ between the EDL and the bulk solution, and of the local ionic concentrations c_+ and c_-:

$$\mathcal{U} = \int_V \left[-\frac{\epsilon_m}{2}|\nabla\phi|^2 + ze(c_+ - c_-)\phi \right] \mathbf{dr}. \tag{2.1}$$

In this expression, z denotes the ion valence, ϵ_m the solvent permittivity, which is assumed for now to remain constant within the EDL, and e the elementary charge.

In the presence of steric crowding effects, the entropic contribution yields [5]

$$-T\mathcal{S} = \int_V \frac{kT}{a^3}\Big[a^3 c_+ \ln(a^3 c_+) + a^3 c_- \ln(a^3 c_-) + \\ [1 - a^3(c_+ - c_-)] \ln[1 - a^3(c_+ + c_-)]\Big]\mathbf{dr}, \tag{2.2}$$

where k denotes the Boltzmann constant and T the ambient temperature. A dimensionless packing parameter ν can be used to account for ionic steric effects in the solution due to the finite size of ions, and can be related to the effective ionic diameter a and to the bulk solution concentration $c_{b,\infty}$ far from the particle through the estimate [4]

$$\nu = 2a^3 c_{b,\infty}. \tag{2.3}$$

Setting the variational derivative $\delta\mathcal{F}/\delta c_\pm$ equal to zero, we obtain an expression for the local ionic concentration

$$c_\pm = \frac{c_b \exp\left(\mp \dfrac{ze\phi}{kT}\right)}{1 + 2\nu \sinh^2 \dfrac{ze\phi}{2kT}} \tag{2.4}$$

where c_b denotes the bulk concentration in the immediate vicinity of the EDL.[1] Then, setting the variational derivative $\delta\mathcal{F}/\delta\phi$ equal to zero, we obtain a modified Poisson-Boltzmann equation that accounts for steric effects

$$\nabla^2\phi = \frac{2zec_b}{\epsilon_m} \frac{\sinh \dfrac{ze\phi}{kT}}{1 + 2\nu \sinh^2 \dfrac{ze\phi}{2kT}} \tag{2.5}$$

where the right-hand side of the equation represents the negative of the local charge density ρ_E. As reviewed by Bazant *et al.* [6], Equation (2.5) was derived for the first time by Bikerman [40] in 1942, and has since been independently rediscovered in several studies. For $\nu = 0$, steric effects are neglected and we recover the classical Poisson-Boltzmann equation.

[1] Note that c_b varies around the surface of the particle and is a function of the polar angle θ.

The present study focuses on large spherical particles whose radius R is orders of magnitude larger than the characteristic thickness of the EDL, which is classically given by the Debye length

$$\lambda_D = \sqrt{\frac{\epsilon_m kT}{2z^2e^2c_{b,\infty}}}. \tag{2.6}$$

For such particles, the EDL is usually modelled as an infinite plane, as shown in Figure 1-1, and the modified Poisson-Boltzmann equation can then be rewritten as

$$\frac{\partial^2\phi}{\partial y^2} = \frac{2zec_b}{\epsilon_m}\frac{\sinh\dfrac{ze\phi}{kT}}{1+2\nu\sinh^2\dfrac{ze\phi}{2kT}}. \tag{2.7}$$

At the surface of the particle, ϕ is equal to the zeta potential ζ, defined here as the potential difference between the particle surface and the bulk electrolyte. Equation (2.7) can be integrated once to yield [4]

$$\frac{\partial\phi}{\partial y} = -\mathrm{sgn}[\zeta]\frac{2zec_b\lambda_D}{\epsilon_m}\sqrt{\frac{2}{\nu}\ln\left(1+2\nu\sinh^2\frac{ze\phi}{2kT}\right)} \tag{2.8}$$

At this point, we can employ dimensionless variables to facilitate the analysis of the physical processes involved. We define the Debye length λ_D as the characteristic length scale and the thermal voltage

$$\varphi_T = \frac{kT}{ze} \tag{2.9}$$

as the characteristic potential scale. With the aforementioned scales, the modified Poisson-Boltzmann equation reduces to

$$\frac{\partial^2\phi}{\partial y^2} = \frac{c_b\sinh\phi}{1+2\nu\sinh^2\dfrac{\phi}{2}}. \tag{2.10}$$

This dimensionless equation can be integrated once to yield [4], again in dimensionless

form,

$$\frac{\partial \phi}{\partial y} = -\text{sgn}[\zeta]\sqrt{\frac{2c_b}{\nu}\ln\left(1 + 2\nu \sinh^2\frac{\phi}{2}\right)}. \tag{2.11}$$

Observe that Equation (2.8) is proportional to the dimensional form of c_b, while Equation (2.11) is proportional to the square root of dimensionless c_b. Equation (2.11) can in turn be integrated numerically to obtain the dimensionless potential distribution across the EDL.

2.1.2 Electro-osmotic and diffusio-osmotic flows

Fluid motion in the EDL is the result of two distinct phenomena: firstly, the co-existence of a charged region in the EDL and of an externally applied electric field leads to electro-osmotic flow; secondly, concentration gradients polarize the cloud of counterions surrounding the particle, resulting in a fluid motion driven by these gradients. The latter is a phenomenon usually referred to as diffusio-osmotic flow [28, 60].

The low Reynolds number flow in the EDL can be described by the Stokes equation. Based on the lubrication approximation, there is no fluid motion in the thickness of the EDL. Hence, the projection of the Stokes equation in the $\mathbf{e_y}$ direction reads

$$-\frac{\partial P}{\partial y} - \rho_E \frac{\partial \phi}{\partial y} = 0. \tag{2.12}$$

Combined with the modified Poisson-Boltzmann equation derived earlier, Equation (2.12) becomes, in dimensional form,

$$\frac{\partial P}{\partial y} - 2zec_b \frac{\sinh\left(\dfrac{ze\phi}{kT}\right)}{1 + 2\nu \sinh^2\left(\dfrac{ze\phi}{2kT}\right)}\frac{\partial \phi}{\partial y} = 0. \tag{2.13}$$

Integration of Equation (2.13) with respect to y yields the osmotic pressure profile

across the EDL:

$$P(y) = P_\infty + \frac{2kTc_b}{\nu} \ln\left[1 + 2\nu \sinh^2\left(\frac{ze\phi}{2kT}\right)\right]. \tag{2.14}$$

The projection of the Stokes equation on $\mathbf{e_x}$ reads

$$-\frac{\partial P}{\partial x} - \rho_E \frac{\partial \phi}{\partial x} + \eta \frac{\partial^2 u}{\partial y^2} = 0. \tag{2.15}$$

Using Equation (2.14) for the pressure found previously, we obtain

$$\eta \frac{\partial^2 u}{\partial y^2} = \frac{2kT}{\nu} \ln\left[1 + 2\nu \sinh^2\left(\frac{ze\phi}{2kT}\right)\right] \frac{\partial c_b}{\partial x} - \epsilon_m \frac{\partial^2 \phi}{\partial y^2} \frac{\partial \psi_b}{\partial x}. \tag{2.16}$$

Interestingly, as noted by Prieve *et al.* [28], the velocity depends only upon the electric field and the ionic concentration in the bulk solution. To obtain the slip velocity of the particle, we can integrate Equation (2.16) twice to obtain

$$u_s = \frac{\epsilon_m \zeta}{\eta} \frac{\partial \psi_b}{\partial x} - \frac{k^2 T^2 \epsilon_m}{\eta z^2 e^2} \kappa(\zeta, 0) \frac{1}{c_b} \frac{\partial c_b}{\partial x}, \tag{2.17}$$

where the coefficient $\kappa(\zeta, \xi)$ is given by[2]

$$\kappa(\zeta, \xi) = \int_\xi^\zeta \int_0^1 \frac{\phi''}{2} \sqrt{\frac{\ln\left(1 + 2\nu \sinh^2 \frac{\phi' \phi''}{2}\right)}{\ln\left(1 + 2\nu \sinh^2 \frac{\phi''}{2}\right)}} d\phi' d\phi''. \tag{2.18}$$

Equation (2.17) describes the slip velocity as a combination of electro-osmotic and diffusio-osmotic slips with consideration given to steric effects, and is an equivalent form of the Dukhin-Derjaguin slip formula [18] for "first kind" electro-osmosis with a thin quasi-equilibrium EDL. Rica and Bazant obtained an equivalent formula for the Gouy-Chapman model in their study of electrodiffusiophoresis [31]. A formulation for the coefficient in Equation (2.18) was first obtained by Kilic [61] as a general function of osmotic pressure. In this study, the coefficient in Equation (2.18) has

[2] ξ is a dummy variable used to denote the lower limit of the outer integral.

been specifically derived for the steric model presented here.

Again, we can consider a dimensionless version of Equation (2.17) by introducing the characteristic velocity

$$U^* = \frac{\epsilon_m \varphi_T^2}{\eta R}. \tag{2.19}$$

With this scaling, we find that the dimensionless slip velocity u_s at the surface of the particle is

$$u_s = \zeta \frac{\partial \psi_b}{\partial x} - \frac{\kappa(\zeta, 0)}{c_b} \frac{\partial c_b}{\partial x}. \tag{2.20}$$

2.2 Bulk solution

2.2.1 Electric potential and ionic concentration

Following most prior work on the electrophoresis of charged spherical particles with thin EDLs, we assume that the bulk solution is electroneutral, so that

$$c_\pm = c_b. \tag{2.21}$$

We also assume that the diffusivities of the cations and the anions are identical. The ion fluxes in the bulk solution are given by the dimensional Nernst-Planck relation

$$\mathbf{j}^\pm = -D\nabla c_b \mp \frac{zeD}{kT} c_b \nabla \Psi_b + c_b \mathbf{u}, \tag{2.22}$$

which describes the flux of each ionic species as a combination of diffusion, electromigration and advection. It is again convenient to nondimensionalize the governing equations of the model to facilitate their analysis. Scaling the ionic flux by the quantity

$$J = \frac{D c_{b,\infty}}{R}, \tag{2.23}$$

we obtain the dimensionless Nernst-Planck relation

$$\mathbf{j}^\pm = -\nabla c_b \mp c_b \nabla \Psi_b + \mathrm{Pe} c_b \mathbf{u}, \tag{2.24}$$

where Pe is a Péclet number which compares advective transport and diffusion and is defined by

$$\mathrm{Pe} = \frac{U^* R}{D} = \frac{\epsilon_m \varphi_T^2}{\eta D}. \tag{2.25}$$

The conservation of counterions and coions are expressed by the relations

$$\begin{cases} \mathrm{div}(\mathbf{j}^+) = -\nabla^2 c_b - \mathrm{div}(c_b \nabla \Psi_b) + \mathrm{Pe}\ \mathrm{div}(c_b \mathbf{u}) = 0, \\ \\ \mathrm{div}(\mathbf{j}^-) = -\nabla^2 c_b + \mathrm{div}(c_b \nabla \Psi_b) + \mathrm{Pe}\ \mathrm{div}(c_b \mathbf{u}) = 0, \end{cases} \tag{2.26}$$

which can be combined to yield, in dimensionless form,

$$\begin{cases} \nabla^2 c_b - \mathrm{Pe}\ \mathrm{div}(c_b \mathbf{u}) = 0, \\ \\ \mathrm{div}(c_b \nabla \Psi_b) = 0. \end{cases} \tag{2.27}$$

These equations describe the concentration and electric potential fields in the bulk solution and are classically referred to as the Poisson-Nernst-Planck equations [3].

The inner and the outer asymptotic solutions are matched by appropriate boundary conditions expressing the conservation of ions across the interface. Chu and Bazant [2] developed a systematic strategy to obtain effective boundary conditions at the interface between the EDL and the bulk solution, demonstrating that excess quantities have to be considered to establish the boundary conditions of conservation relations. Here, the excess ion fluxes in the electric double layer are defined as the difference between ion fluxes in the EDL and the ion fluxes in the bulk solution:

$$\mathbf{J}^{\pm}_{\mathbf{exc}} = -\frac{D}{kT}(c_{\pm} - c_b)\nabla \mu_b + (c_{\pm} u - c_b u_s), \tag{2.28}$$

where $\mu_b = kT \ln c_b \pm ze\Psi_b$ denotes the electrochemical potential in the bulk solution,

and $u = u(y)$ is given by

$$u(y) = \frac{\epsilon_m(\zeta - \phi(y))}{\eta}\frac{\partial \psi_b}{\partial x} - \frac{k^2T^2\epsilon_m}{\eta z^2 e^2}\kappa(\zeta, \phi(y))\frac{1}{c_b}\frac{\partial c_b}{\partial x}. \tag{2.29}$$

The total excess surface fluxes can be calculated by integrating Equation (2.28) between $y = 0$ and $+\infty$ in the EDL to yield, in outer spherical coordinates,

$$\mathbf{J}_{\mathbf{S}}^{\pm} = \int_0^{+\infty} \mathbf{J}_{\mathbf{exc}}^{\pm}(y)dy = -\frac{D}{kT}c_b\Gamma_{\pm}\nabla_S\mu_b + \int_0^{+\infty}(c_{\pm}u - c_b u_s)dy. \tag{2.30}$$

In this relation, $\Gamma_{\pm}$ denotes the surface adsorption coefficient [2], defined by

$$\Gamma_{\pm} = \frac{1}{c_b}\int_0^{+\infty}(c_{\pm}(y) - c_b)dy, \tag{2.31}$$

and ∇_S denotes the surface gradient operator in the outer spherical coordinate system. The equation of ionic transport normal to the boundary is

$$j_n^{\pm} = -\frac{D}{kT}c_b\frac{\partial \mu_b}{\partial r}. \tag{2.32}$$

Finally, the conservation for ionic species at the interface of the EDL reads

$$j_n^{\pm} + \mathrm{div}_S\mathbf{J}_{\mathbf{S}}^{\pm} = 0, \tag{2.33}$$

so that

$$\begin{cases} j_n^+ + j_n^- = -\mathrm{div}_S(\mathbf{J}_{\mathbf{S}}^+ + \mathbf{J}_{\mathbf{S}}^-), \\ \\ j_n^+ - j_n^- = -\mathrm{div}_S(\mathbf{J}_{\mathbf{S}}^+ - \mathbf{J}_{\mathbf{S}}^-), \end{cases} \tag{2.34}$$

where div_S denotes the surface divergence operator. In outer spherical coordinates,

these relations become

$$\begin{cases} \dfrac{\partial c_b}{\partial n} = -\dfrac{1}{2}\mathrm{div}_S\left((w + p_1\mathrm{Pe})\nabla_S \ln c_b + (q + p_2\mathrm{Pe})\dfrac{ze}{kT}\nabla_S\Psi_b\right), \\ \\ \dfrac{zec_b}{kT}\dfrac{\partial \Psi_b}{\partial n} = -\dfrac{1}{2}\mathrm{div}_S\left((q + p_3\mathrm{Pe})\nabla_S \ln c_b + (w + p_4\mathrm{Pe})\dfrac{ze}{kT}\nabla_S\Psi_b\right), \end{cases} \tag{2.35}$$

where q denotes the excess charge stored in the EDL, w the excess ion concentration, and p_i quantities related to the excess advective flux. These quantities are defined respectively by

$$q = \int_0^{+\infty} \Big[c_+(y) - c_-(y)\Big]dy = \int_0^{+\infty} \rho_E(y)dy, \tag{2.36}$$

$$w = \int_0^{+\infty} \Big[c_+(y) + c_-(y) - 2c_b\Big]dy, \tag{2.37}$$

$$p_1 = \int_0^{+\infty} \Big[\{c_+(y) + c_-(y)\}\,\kappa(\zeta, \phi(y)) - 2c_b\kappa(\zeta, 0)\Big]dy, \tag{2.38}$$

$$p_2 = \int_0^{+\infty} -\Big[\{c_+(y) + c_-(y)\}\,(\zeta - \phi(y)) - 2c_b\zeta\Big]dy, \tag{2.39}$$

$$p_3 = \int_0^{+\infty} \Big[\{c_+(y) - c_-(y)\}\,\kappa(\zeta, \phi(y))\Big]dy, \tag{2.40}$$

$$p_4 = \int_0^{+\infty} -\Big[\{c_+(y) - c_-(y)\}\,(\zeta - \phi(y))\Big]dy. \tag{2.41}$$

The excess ion concentration can be related to the Dukhin-Bikerman number, Du, through the relation

$$\mathrm{Du} = \frac{w}{Rc_b}. \tag{2.42}$$

With the steric model developed by Kilic and Bazant [4, 5] and described in Section 2.1.1, the excess charge q stored in the EDL is given as a function of the zeta potential ζ by

$$q = -2zec_{b,\infty}\lambda_D \mathrm{sgn}(\zeta)\sqrt{\frac{2c_b}{\nu}\ln\left[1 + 2\nu\sinh^2\left(\frac{ze\zeta}{2kT}\right)\right]}. \tag{2.43}$$

The excess ion concentration is defined by the integral expression [4, 5]

$$w = \int_0^{\zeta} \frac{2c_{b,\infty}\lambda_D\sqrt{c_b}(1-\nu)}{\sqrt{\frac{2}{\nu}\ln(1+2\nu\sinh^2(\frac{u}{2}))}} \frac{(\cosh u - 1)du}{1+2\nu\sinh^2(\frac{u}{2})}, \tag{2.44}$$

which can be evaluated numerically. Likewise, the advective quantities p_i can be evaluated numerically using the following integral expressions

$$p_1 = \int_0^{\zeta} \frac{2c_{b,\infty}\lambda_D\sqrt{c_b}}{\sqrt{\frac{2}{\nu}\ln(1+2\nu\sinh^2(\frac{u}{2}))}} \left(\frac{\cosh u}{1+2\nu\sinh^2(\frac{u}{2})}\kappa(\zeta,u) - \kappa(\zeta,0) \right) du, \tag{2.45}$$

$$p_2 = -\int_0^{\zeta} \frac{2c_{b,\infty}\lambda_D\sqrt{c_b}}{\sqrt{\frac{2}{\nu}\ln(1+2\nu\sinh^2(\frac{u}{2}))}} \left(\frac{\cosh u}{1+2\nu\sinh^2(\frac{u}{2})}(\zeta-u) - \zeta \right) du, \tag{2.46}$$

$$p_3 = -\int_0^{\zeta} \frac{2c_{b,\infty}\lambda_D\sqrt{c_b}}{\sqrt{\frac{2}{\nu}\ln(1+2\nu\sinh^2(\frac{u}{2}))}} \left(\frac{\sinh u}{1+2\nu\sinh^2(\frac{u}{2})}\kappa(\zeta,u) \right) du, \tag{2.47}$$

$$p_4 = \int_0^{\zeta} \frac{2c_{b,\infty}\lambda_D\sqrt{c_b}}{\sqrt{\frac{2}{\nu}\ln(1+2\nu\sinh^2(\frac{u}{2}))}} \left(\frac{\sinh u}{1+2\nu\sinh^2(\frac{u}{2})}(\zeta-u) \right) du. \tag{2.48}$$

Scaling the zeta potential by the thermal voltage and the excess charge stored in the EDL by the quantity

$$q^* = zec_{b,\infty}\lambda_D, \tag{2.49}$$

we obtain the dimensionless expression

$$q = -2\mathrm{sgn}(\zeta)\sqrt{\frac{2c_b}{\nu}\ln\left[1+2\nu\sinh^2\left(\frac{\zeta}{2}\right)\right]}. \tag{2.50}$$

Similarly, scaling the zeta potential by the thermal voltage and the excess ionic concentration in the EDL by the quantity

$$w^* = c_{b,\infty}\lambda_D, \tag{2.51}$$

we find

$$w = \int_0^{\zeta} 2\sqrt{c_b}\frac{\cosh u - 1}{1 + 2\nu\sinh^2(\frac{u}{2})}\frac{(1-\nu)du}{\sqrt{\frac{2}{\nu}\ln(1 + 2\nu\sinh^2(\frac{u}{2}))}}. \tag{2.52}$$

Finally, by performing a similar scaling for p_i, we obtain

$$p_1 = \int_0^{\zeta} \frac{2\sqrt{c_b}}{\sqrt{\frac{2}{\nu}\ln(1 + 2\nu\sinh^2(\frac{u}{2}))}}\left(\frac{\cosh u}{1 + 2\nu\sinh^2(\frac{u}{2})}\kappa(\zeta, u) - \kappa(\zeta, 0)\right)du, \tag{2.53}$$

$$p_2 = -\int_0^{\zeta} \frac{2\sqrt{c_b}}{\sqrt{\frac{2}{\nu}\ln(1 + 2\nu\sinh^2(\frac{u}{2}))}}\left(\frac{\cosh u}{1 + 2\nu\sinh^2(\frac{u}{2})}(\zeta - u) - \zeta\right)du, \tag{2.54}$$

$$p_3 = -\int_0^{\zeta} \frac{2\sqrt{c_b}}{\sqrt{\frac{2}{\nu}\ln(1 + 2\nu\sinh^2(\frac{u}{2}))}}\left(\frac{\sinh u}{1 + 2\nu\sinh^2(\frac{u}{2})}\kappa(\zeta, u)\right)du, \tag{2.55}$$

$$p_4 = \int_0^{\zeta} \frac{2\sqrt{c_b}}{\sqrt{\frac{2}{\nu}\ln(1 + 2\nu\sinh^2(\frac{u}{2}))}}\left(\frac{\sinh u}{1 + 2\nu\sinh^2(\frac{u}{2})}(\zeta - u)\right)du. \tag{2.56}$$

The dimensionless form of the boundary conditions (2.35) then becomes

$$\begin{cases} \dfrac{\partial c_b}{\partial n} = -\dfrac{\epsilon}{2}\mathrm{div}_S\Big((w + p_1\mathrm{Pe})\nabla_S \ln c_b + (q + p_2\mathrm{Pe})\nabla_S\Psi_b\Big), \\ \\ c_b\dfrac{\partial \Psi_b}{\partial n} = -\dfrac{\epsilon}{2}\mathrm{div}_S\Big((q + p_3\mathrm{Pe})\nabla_S \ln c_b + (w + p_4\mathrm{Pe})\nabla_S\Psi_b\Big), \end{cases} \tag{2.57}$$

where ϵ denotes the ratio of the Debye length to the radius of the particle. These boundary conditions state that fluxes of ions are transported across the EDL/bulk interface to balance the tangential surface flux gradients in the EDL.

2.2.2 Zeta potential determination

The zeta potential is defined as the electric potential difference across the EDL

$$\zeta(\theta) = \Psi_P - \psi_b(\theta), \tag{2.58}$$

where the potential Ψ_P of the particle is initially unknown.[3] To evaluate the electric potential of the particle, as demonstrated by Yariv [39], we have to rely on global charge conservation on the surface of the particle. The total charge Q of the particle remains invariant during the formation of the electric double layer. Hence,

$$Q = -\int_S q dA = -2\pi R^2 \int_0^\pi q(\theta) \sin\theta d\theta, \tag{2.59}$$

where q is the local surface charge and is related to the zeta potential through Equation (2.43), when the steric model developed by Kilic *et al.* [4, 5] is employed to describe the physics of the EDL. In dimensionless form, we obtain

$$Q = \int_0^\pi \text{sign}(\zeta) \sqrt{\frac{2c_b}{\nu} \ln\left[1 + 2\nu \sinh^2\left(\frac{\zeta}{2}\right)\right]} \sin\theta d\theta, \tag{2.60}$$

where the global charge Q of the particle has been scaled by the quantity

$$Q^* = 4\pi R^2 z e c_{b,\infty} \lambda_D. \tag{2.61}$$

2.2.3 Velocity field and electrophoretic velocity

As evidenced by Equation (2.27), at high Péclet numbers, advective transport introduces coupling between the ionic concentration and velocity fields in the bulk solution. Thus, the calculation of the electrophoretic mobility can only be achieved through a complete determination of the velocity field everywhere. In the bulk solution, the velocity is the solution of the Stokes equation[4]

$$\eta \nabla^2 \mathbf{u} - \nabla P - \epsilon_m \nabla^2 \Psi_b \nabla \Psi_b = \mathbf{0}, \tag{2.62}$$

[3] Due to the axisymmetry of the problem, the model equations can be reduced to two dimensions, and hence ψ_b is expressed as a function of only θ.

[4] This derivation is done in the frame where the particle is stationary. The electric body force acts in the $-\mathbf{e_z}$ direction and thus generally in the $\mathbf{e}_\theta$ direction, meaning that when one transforms back into the lab frame, the particle is moving at a velocity $-U\mathbf{e_z}$. As we will see later, this means that in the frame of the particle, the fluid adopts a velocity $U\mathbf{e_z}$ at infinity.

where $\mathbf{u}$ denotes the fluid velocity, P the pressure field, ϵ_m the solvent permittivity and Ψ_b the electric potential in the bulk electrolyte. The Stokes equation is considered along with the continuity equation

$$\nabla \cdot \mathbf{u} = 0. \tag{2.63}$$

The problem considered here is very similar to the one of a sphere in a Stokes flow, with the notable exceptions that slip boundary conditions have to be considered at the surface of the sphere and that an external force appears in the expression of the Stokes equation. The field-induced variations of the bulk concentration result in corresponding variations of the solution potential, which in turn result in a body force acting on the bulk fluid.

The classical resolution of the problem relies on the vorticity, defined by $\omega = \nabla \times \mathbf{u}$. For the bi-dimensional flow considered here, the vorticity is directed along the basis vector e_ψ in spherical coordinates. Using the continuity equation, we find that

$$\nabla \times \omega = \nabla \times \nabla \times \mathbf{u} = -\nabla^2 \mathbf{u}. \tag{2.64}$$

The momentum conservation can then be reformulated in terms of the vorticity to be

$$-\eta \nabla \times \omega - \nabla P - \epsilon_m \nabla^2 \Psi_b \nabla \Psi_b = \mathbf{0}. \tag{2.65}$$

Taking the curl of this relation, we finally obtain

$$\nabla \times \eta \nabla \times \omega + \nabla \times \epsilon_m \nabla^2 \Psi_b \nabla \Psi_b = \mathbf{0}. \tag{2.66}$$

At this point, we introduce the Stokes stream function Ψ, defined in spherical coordinates by

$$u_r = \frac{1}{r^2 \sin\theta} \frac{\partial \Psi}{\partial \theta}, \quad u_\theta = -\frac{1}{r \sin\theta} \frac{\partial \Psi}{\partial r}. \tag{2.67}$$

At infinity, the velocity is $\mathbf{u} = U\mathbf{e_z}$. This means that

$$\Psi(r,\theta) \sim \frac{U}{2} r^2 \sin^2\theta \tag{2.68}$$

when $r \to +\infty$. It is thus convenient to rewrite the Stokes stream function in the form

$$\Psi(r,\theta) = \frac{r^2}{2}\sin^2(\theta)\Psi_S(r,\theta) \tag{2.69}$$

for some function Ψ_S. The boundary conditions of this function at infinity are given by

$$\Psi_S(r,\theta) \to U, \qquad r \to +\infty, \tag{2.70}$$

and

$$\frac{\partial \Psi_S}{\partial r}(r,\theta) \to 0, \qquad r \to +\infty. \tag{2.71}$$

At the surface of the sphere, the component of the velocity normal to the surface vanishes, while the component tangential to the surface is equal to the slip velocity. Using Equation (2.67), the components of the velocity can be expressed in terms of Ψ_S as

$$\begin{cases} u_r = \Psi_S(r,\theta)\cos\theta + \dfrac{\sin\theta}{2}\dfrac{\partial \Psi_S}{\partial \theta}(r,\theta), \\ \\ u_\theta = -\Psi_S(r,\theta)\sin\theta - \dfrac{r\sin\theta}{2}\dfrac{\partial \Psi_S}{\partial r}(r,\theta). \end{cases} \tag{2.72}$$

Thus, the boundary conditions at the surface are given by

$$\begin{cases} \Psi_S(r,\theta)\Big|_{r=1} = 0, \\ \\ -\dfrac{r\sin\theta}{2}\dfrac{\partial \Psi_S}{\partial r}(r,\theta)\Big|_{r=1} = u_{slip}. \end{cases} \tag{2.73}$$

The vorticity can be expressed as a function of Ψ_S through the relation

$$\omega = -\frac{1}{r\sin\theta}\mathcal{L}\Psi_S \mathbf{e}_\psi, \tag{2.74}$$

where operator $\mathcal{L}$ is defined by

$$\mathcal{L} = \frac{r^2 \sin^2\theta}{2}\left[\frac{\partial^2.}{\partial r^2} + \frac{4}{r}\frac{\partial.}{\partial r} + \frac{1}{r^2}\frac{\partial^2.}{\partial \theta^2} + \frac{3\cot\theta}{r^2}\frac{\partial.}{\partial \theta}\right]. \tag{2.75}$$

The curl of the vorticity is then given by

$$\nabla \times \Psi_S = -\frac{1}{r^2 \sin\theta}\frac{\partial \mathcal{L}\Psi_S}{\partial \theta}\mathbf{e_r} + \frac{1}{r\sin\theta}\frac{\partial \mathcal{L}\Psi_S}{\partial r}\mathbf{e}_\theta. \tag{2.76}$$

Taking the curl again on both sides, we finally obtain

$$\nabla \times \nabla \times \Psi_S = \frac{1}{r\sin\theta}\mathcal{E}\mathcal{L}\Psi_S \tag{2.77}$$

where the operator $\mathcal{E}$ is defined as

$$\mathcal{E} = \frac{\partial^2.}{\partial r^2} + \frac{\sin\theta}{r^2}\frac{\partial}{\partial \theta}\left(\frac{1}{\sin\theta}\frac{\partial.}{\partial \theta}\right). \tag{2.78}$$

The derivation above gives the slip velocity $\mathbf{u_s}$ around a charged particle in a fluid that has a velocity $\mathbf{U} = U\mathbf{e_z}$ at infinity. This is an analysis performed in a reference frame in which the particle is stationary.[5] In our problem of interest, the fluid is stationary at infinity and our particle is moving at a velocity $\mathbf{U}$. This means that the appropriate velocity boundary condition to be taken at the surface of the particle should be $\mathbf{u} = \mathbf{U} - \mathbf{u_s}$. We can consider the problem of interest a superposition of two problems, since the Stokes equation is linear: Problem 1, in which we consider pure Stokes flow around a sphere in a reference frame such that the fluid is at rest at infinity and the sphere is moving at a velocity $\mathbf{U}$, and Problem 2, which is a superposition of the opposite of the flow we considered in the derivation above and a pure Stokesian flow such that the fluid is at rest at infinity and the slip velocity at the surface of the sphere is $-\mathbf{u_s}$. Let us list the equations and boundary conditions more explicitly for clarity.

[5]By transforming to the lab frame, we see that the fluid is now at rest and the slip velocity is given by $\mathbf{u_s} - \mathbf{U}$, with the particle moving at a velocity $-\mathbf{U}$. Recall that in our definition, the electric body force is acting in the $-\mathbf{e_z}$ direction. This is consistent with the direction of motion of the particle.

For our problem of interest, which is a superposition of Problems 1 and 2, the Stokes equation reads

$$\nabla \cdot \sigma_H + \nabla \cdot \sigma_E = 0 \tag{2.79}$$

where σ_E denotes the electrostatic Maxwell stress tensor and σ_H is the hydrodynamic constraint tensor, which reads

$$\sigma_H = -p\mathbf{I} + (\nabla \mathbf{u} + \nabla \mathbf{u}^T) \tag{2.80}$$

where p is the local hydrodynamic pressure. At infinity, the velocity field vanishes:

$$\mathbf{u}|_{r\to\infty} = 0, \tag{2.81}$$

while at the surface of the particle, the velocity field is

$$\mathbf{u}|_{r=R} = U\mathbf{e_z} - u_s\mathbf{e}_\theta. \tag{2.82}$$

For Problem 1, we consider a pure Stokes flow

$$\nabla \cdot \sigma_H^{(1)} = 0 \tag{2.83}$$

where $\sigma_H^{(1)}$ denotes the hydrodynamic constraint tensor for Problem 1

$$\sigma_H^{(1)} = -p^{(1)}\mathbf{I} + (\nabla \mathbf{u}^{(1)} + \nabla \mathbf{u}^{(1)T}). \tag{2.84}$$

The flow boundary conditions for Problem 1 are

$$\mathbf{u}^{(1)}|_{r\to\infty} = 0 \tag{2.85}$$

and

$$\mathbf{u}^{(1)}|_{r=R} = U\mathbf{e_z}. \tag{2.86}$$

In Problem 2, we consider the superposition of two flows. The first is the flow

we considered in the derivation above with a reversal in axes such that the flow at infinity is $-U\mathbf{e_z}$ and the slip velocity is $-u_s\mathbf{e}_\theta$. The second is a pure Stokes flow in which the fluid has a velocity $U\mathbf{e_z}$ at infinity and the particle is at rest. This gives us the following governing equation

$$\nabla \cdot \sigma_H^{(2)} + \nabla \cdot \sigma_E = 0 \tag{2.87}$$

where σ_H is the hydrodynamic constraint tensor for Problem 2, which reads

$$\sigma_H^{(2)} = -p^{(2)}\mathbf{I} + (\nabla \mathbf{u}^{(2)} + \nabla \mathbf{u}^{(2)T}). \tag{2.88}$$

The flow boundary conditions for Problem 2 are

$$\mathbf{u}^{(2)}|_{r\to\infty} = 0 \tag{2.89}$$

and

$$\mathbf{u}^{(2)}|_{r=R} = -u_s\mathbf{e}_\theta. \tag{2.90}$$

We now use the Lorentz reciprocal theorem used earlier by Stone and Samuel [62] and by Anderson [9] in similar situations, except that we modify it to allow for the presence of a non-hydrodynamic stress tensor:

$$\int_V \mathbf{u}^{(1)} \cdot \nabla \cdot \sigma_E \; dV + \int_S \mathbf{u}^{(1)} \cdot \sigma_H^{(2)} \cdot \mathbf{n} \; dS = \int_S \mathbf{u}^{(2)} \cdot \sigma_H^{(1)} \cdot \mathbf{n} \; dS \tag{2.91}$$

where S can be taken to be the particle surface, noting that the velocity fields for both Problems 1 and 2 decay to zero at infinity.

The solution for Problem 1 is known. We have

$$u_r^{(1)} = \frac{U\cos\theta}{2}\left(\frac{3R}{r} - \frac{R^3}{r^3}\right) \tag{2.92}$$

and

$$u_\theta^{(2)} = -\frac{U\sin\theta}{4}\left(\frac{3R}{r} + \frac{R^3}{r^3}\right). \tag{2.93}$$

Noting that $\nabla p^{(1)} = \eta \nabla^2 \mathbf{u}^{(1)}$, this gives us a pressure field

$$p^{(1)} = \frac{3}{2} RU\eta \frac{\cos\theta}{r^2}. \tag{2.94}$$

Noting, also, that the hydrodynamic constraint tensor components are given by

$$\sigma_{rr}^{(1)} = -p^{(1)} + 2\eta \frac{\partial u_r^{(1)}}{\partial r} \tag{2.95}$$

and

$$\sigma_{r\theta}^{(1)} = \eta \left[r \frac{\partial}{\partial r} \left(\frac{u_\theta^{(1)}}{r} \right) + \frac{1}{r} \frac{\partial u_r^{(1)}}{\partial \theta} \right], \tag{2.96}$$

we obtain, at the surface of the particle,

$$\sigma_H^{(1)} \cdot \mathbf{n} = \frac{3\eta U}{2R} \mathbf{e_z}, \tag{2.97}$$

where $\sigma_H^{(1)}$ gives an indication of the hydrodynamic drag force on the fluid due to the particle's motion.[6] This allows us to rewrite the Lorentz theorem (2.91) as

$$U\mathbf{e_z} \cdot \int_S \sigma_H^{(2)} \cdot \mathbf{n} \; dS = \frac{3\eta U}{2R} \mathbf{e_z} \cdot \int_S -\mathbf{u_s} \; dS - \int_V \mathbf{u}^{(1)} \cdot \nabla \cdot \sigma_E \; dV. \tag{2.98}$$

Since our problem involves freely suspended particles, we invoke the constraint that there is no net force on the particle in our problem of interest, thus giving

$$-\mathbf{e_z} \cdot \left(\int_S \left(\sigma_H^{(1)} + \sigma_H^{(2)} + \sigma_E \right) \cdot \mathbf{n} \; dS \right) = 0. \tag{2.99}$$

Noting that $\mathbf{e_z} \cdot \int_S \sigma_H^{(1)} \cdot \mathbf{n} \; dS = 6\eta\pi RU$, this gives us an expression for the electrophoretic speed U

$$U = \mathbf{e_z} \cdot \frac{1}{4\pi R^2} \int_S \mathbf{u_s} \, dS - \frac{1}{6\eta\pi R} \left(\mathbf{e_z} \cdot \int_S \sigma_E \cdot \mathbf{n} \; dS - \frac{1}{U} \int_V \mathbf{u}^{(1)} \cdot \nabla \cdot \sigma_E \; dV \right). \tag{2.100}$$

In the absence of an electric field, $\mathbf{u_s} = 0$ everywhere and thus $U = 0$. In the weak-field

[6] Note that the net stress tensor on the particle is of the opposite sign, as expected from Newton's Third Law, and acts to oppose the particle's forward motion.

limit, we can neglect σ_E, giving us $U = \mathbf{e_z} \cdot \frac{1}{4\pi R^2} \int_S \mathbf{u_s}\ dS$.[7]

The expression in parentheses in the equation above can be simplified further by noting that

$$\nabla \cdot (\sigma_E \cdot \mathbf{u}^{(1)}) = \mathbf{u}^{(1)} \cdot \nabla \cdot \sigma_E + \mathrm{Tr}(\sigma_E \nabla \mathbf{u}^{(1)}). \tag{2.101}$$

Hence,

$$\frac{1}{U}\int_V \mathbf{u}^{(1)} \cdot \nabla \cdot \sigma_E\ dV = \frac{1}{U}\int_V \nabla \cdot (\sigma_E \cdot \mathbf{u}^{(1)})\ dV - \frac{1}{U}\int_V \mathrm{Tr}(\sigma_E \nabla \mathbf{u}^{(1)})\ dV. \tag{2.102}$$

Using Stokes' theorem, and noting that the velocity field for Problem 1 decays to zero at infinity, we can rewrite the above expression as

$$\frac{1}{U}\int_V \mathbf{u}^{(1)} \cdot \nabla \cdot \sigma_E\ dV = \frac{1}{U}\int_S \sigma_E \cdot \mathbf{u}^{(1)} \cdot \mathbf{n}\ dS - \frac{1}{U}\int_V \mathrm{Tr}(\sigma_E \nabla \mathbf{u}^{(1)})\ dV. \tag{2.103}$$

The expression for the electrophoretic speed U can thus be simplified as follows:

$$U = \mathbf{e_z} \cdot \frac{1}{4\pi R^2}\int_S \mathbf{u_s}\ dS - \frac{1}{6\eta\pi RU}\int_V \mathrm{Tr}(\sigma_E \nabla \mathbf{u}^{(1)})\ dV. \tag{2.104}$$

The Maxwell stress tensor σ_E can be written as

$$\sigma_E = \begin{bmatrix} \frac{1}{2}E_r^2 - \frac{1}{2}E_\theta^2 & E_r E_\theta \\ E_r E_\theta & \frac{1}{2}E_\theta^2 - \frac{1}{2}E_r^2 \end{bmatrix} \tag{2.105}$$

where $E_r = \dfrac{\partial \Psi_b}{\partial r}$ and $E_\theta = \dfrac{1}{r}\dfrac{\partial \Psi_b}{\partial \theta}$. The gradient of the velocity vector $\nabla \mathbf{u}^{(1)}$ can be written as

$$\nabla \mathbf{u}^{(1)} = \mathbf{u}^{(1)} \otimes \nabla = \begin{bmatrix} \dfrac{\partial u_r^{(1)}}{\partial r} & \dfrac{1}{r}\dfrac{\partial u_r^{(1)}}{\partial \theta} - \dfrac{u_\theta^{(1)}}{r} \\ \dfrac{\partial u_\theta^{(1)}}{\partial r} & \dfrac{1}{r}\dfrac{\partial u_\theta^{(1)}}{\partial \theta} + \dfrac{u_r^{(1)}}{r} \end{bmatrix} \tag{2.106}$$

[7] This equation differs by a sign from the expressions obtained by Stone and Samuel [62] and Anderson [9] because they define the perturbation/slip velocities in their expressions in the opposite direction.

where the relevant derivatives of $u_r^{(1)}$ are

$$\frac{\partial u_r^{(1)}}{\partial r} = \frac{U\cos\theta}{2}\left(-\frac{3R}{r^2}+\frac{3R^3}{r^4}\right), \tag{2.107}$$

$$\frac{1}{r}\frac{\partial u_r^{(1)}}{\partial \theta} = -\frac{U\sin\theta}{2}\left(\frac{3R}{r^2}-\frac{R^3}{r^4}\right), \tag{2.108}$$

$$\frac{u_r^{(1)}}{r} = \frac{U\cos\theta}{2}\left(\frac{3R}{r^2}-\frac{R^3}{r^4}\right), \tag{2.109}$$

and the relevant derivatives of $u_\theta^{(1)}$ are

$$\frac{\partial u_\theta^{(1)}}{\partial r} = -\frac{U\sin\theta}{4}\left(-\frac{3R}{r^2}-\frac{3R^3}{r^4}\right), \tag{2.110}$$

$$\frac{1}{r}\frac{\partial u_\theta^{(1)}}{\partial \theta} = -\frac{U\cos\theta}{4}\left(\frac{3R}{r^2}+\frac{R^3}{r^4}\right), \tag{2.111}$$

$$\frac{u_\theta^{(1)}}{r} = -\frac{U\sin\theta}{4}\left(\frac{3R}{r^2}+\frac{R^3}{r^4}\right). \tag{2.112}$$

After some algebra, we find that the trace of the matrix product $\sigma_E\nabla\mathbf{u}^{(1)}$ is

$$\begin{aligned}\mathrm{Tr}(\sigma_E\nabla\mathbf{u}^{(1)}) = &\left(\frac{1}{2}\left(\frac{\partial\psi}{\partial r}\right)^2-\frac{1}{2}\left(\frac{1}{r}\frac{\partial\psi}{\partial\theta}\right)^2\right)U\cos\theta\left(-\frac{9R}{4r^2}+\frac{9R^3}{4r^4}\right)\\ &+\frac{\partial\psi}{\partial r}\frac{1}{r}\frac{\partial\psi}{\partial\theta}U\sin\theta\left(\frac{3R^3}{2r^4}\right).\end{aligned} \tag{2.113}$$

Then, the final expression for U is given by

$$\begin{aligned}U = \mathbf{e_z}\cdot\frac{1}{4\pi R^2}\int_S \mathbf{u_s}\,dS&\\ -\frac{1}{6\eta\pi}\int_V\Bigg[&\left(\frac{1}{2}\left(\frac{\partial\psi}{\partial r}\right)^2-\frac{1}{2}\left(\frac{1}{r}\frac{\partial\psi}{\partial\theta}\right)^2\right)\cos\theta\left(-\frac{9}{4r^2}+\frac{9R^2}{4r^4}\right)\\ &+\frac{\partial\psi}{\partial r}\frac{1}{r}\frac{\partial\psi}{\partial\theta}\sin\theta\left(\frac{3R^2}{2r^4}\right)\Bigg]\,dV.\end{aligned} \tag{2.114}$$

Chapter 3

Dielectric decrement effects and model simplifications

3.1 Dielectric decrement modifications to the original model

When dielectric decrement effects are added to the model developed in Chapter 2, the basic theoretical framework can remain largely intact, but the expressions for some of the physical quantities involved have to be modified. These modifications result from the adoption of Equation (1.2) describing the concentration dependence of the dielectric permittivity, which is restated here for easier reference:

$$\epsilon_m = [\epsilon_w - \alpha(c_+ + c_-)]\epsilon_0. \tag{1.2}$$

We will now list the modifications required, first for the quantities pertaining to the EDL, and then for the quantities pertaining to the bulk solution. The resulting quantities apply to a model that takes into account both steric and dielectric decrement effects.

3.1.1 Electric double layer: Poisson-Boltzmann equation

Recall that the expression for the electrostatic energy $\mathcal{U}$ given in Equation (2.1) is dependent on the dielectric permittivity ϵ_m:

$$\mathcal{U} = \int_V \left[-\frac{\epsilon_m}{2}|\nabla\phi|^2 + ze(c_+ - c_-)\phi \right] \mathbf{dr}. \tag{2.1}$$

This gives rise to a modified expression for the ionic concentration $c_\pm$ in the EDL when we set the variational derivative $\delta\mathcal{F}/\delta c_\pm$ to zero:

$$c_\pm = \frac{c_b \exp\left(\mp\dfrac{ze\phi}{kT} - \dfrac{\epsilon_0\alpha}{2kT}|\nabla\phi|^2\right)}{1 + \nu\left[\cosh\left(\dfrac{ze\phi}{kT}\right)\exp\left(-\dfrac{\epsilon_0\alpha}{2kT}|\nabla\phi|^2\right) - 1\right]}. \tag{3.1}$$

Subsequently, we can set the variational derivative $\delta\mathcal{F}/\delta\phi$ to zero to obtain a modified Poisson-Boltzmann relation in a similar form to that obtained by Hatlo *et al.* [1]:

$$\nabla\cdot(\epsilon_m\nabla\phi) = \frac{2zec_b\sinh\left(\dfrac{ze\phi}{kT}\right)\exp\left(-\dfrac{\epsilon_0\alpha}{2kT}|\nabla\phi|^2\right)}{1 + \nu\left[\cosh\left(\dfrac{ze\phi}{kT}\right)\exp\left(-\dfrac{\epsilon_0\alpha}{2kT}|\nabla\phi|^2\right) - 1\right]}. \tag{3.2}$$

These two expressions can be used in place of Equations (2.4) and (2.5) respectively. For $\nu = 0$ and $\alpha = 0$, both steric and dielectric decrement effects are neglected, and we recover the classical Poisson-Boltzmann equation.

Again, for sufficiently large particles with sufficiently thin double layers, the modified Poisson-Boltzmann relation can be reduced to its one-dimensional form:

$$\frac{\partial}{\partial y}\left(\epsilon_m\frac{\partial\phi}{\partial y}\right) = \frac{2zec_b\sinh\left(\dfrac{ze\phi}{kT}\right)\exp\left(-\dfrac{\epsilon_0\alpha}{2kT}\left|\dfrac{\partial\phi}{\partial y}\right|^2\right)}{1 + \nu\left[\cosh\left(\dfrac{ze\phi}{kT}\right)\exp\left(-\dfrac{\epsilon_0\alpha}{2kT}\left|\dfrac{\partial\phi}{\partial y}\right|^2\right) - 1\right]}. \tag{3.3}$$

This expression can be used in place of its equivalent expression in the steric-only model, Equation (2.7).

Using the Debye length λ_D as the characteristic length, the thermal voltage φ_T as the characteristic potential, $c_{b,\infty}$ as the characteristic concentration and the following quantity

$$\alpha^* = \frac{\alpha c_{b,\infty}}{\epsilon_w} \tag{3.4}$$

as a scaling for the constant α, the nondimensional version of the modified Poisson-Boltzmann relation can be written as

$$\frac{\partial^2\phi}{\partial y^2} - 2\alpha\frac{\partial}{\partial y}\left(\frac{c_b\cosh(\phi)\exp\left(-\alpha\left|\frac{\partial\phi}{\partial y}\right|^2\right)}{1+\nu\left[\cosh(\phi)\exp\left(-\alpha\left|\frac{\partial\phi}{\partial y}\right|^2\right)-1\right]}\frac{\partial\phi}{\partial y}\right) = \frac{c_b\sinh(\phi)\exp\left(-\alpha\left|\frac{\partial\phi}{\partial y}\right|^2\right)}{1+\nu\left[\cosh(\phi)\exp\left(-\alpha\left|\frac{\partial\phi}{\partial y}\right|^2\right)-1\right]}. \tag{3.5}$$

This equation replaces its steric-only equivalent, Equation (2.10).

3.1.2 Electric double layer: Electro-osmotic and diffusio-osmotic flows

Since the expression for the local charge density ρ_E changes with the addition of dielectric decrement effects, the projection of the Stokes equation on $\mathbf{e_y}$ has to be rewritten as

$$\frac{\partial P}{\partial y} = 2zec_b\frac{\sinh\left(\frac{ze\phi}{kT}\right)\exp\left(-\frac{\epsilon_0\alpha}{2kT}\left|\frac{\partial\phi}{\partial y}\right|^2\right)}{1+\nu\left[\cosh\left(\frac{ze\phi}{kT}\right)\exp\left(-\frac{\epsilon_0\alpha}{2kT}\left|\frac{\partial\phi}{\partial y}\right|^2\right)-1\right]}\frac{\partial\phi}{\partial y}. \tag{3.6}$$

Hence, the osmotic pressure profile in the double layer reads

$$P(y^*) = P_\infty + \int_\infty^{y^*} 2zec_b\frac{\sinh\left(\frac{ze\phi}{kT}\right)\exp\left(-\frac{\epsilon_0\alpha}{2kT}\left|\frac{\partial\phi}{\partial y}\right|^2\right)}{1+\nu\left[\cosh\left(\frac{ze\phi}{kT}\right)\exp\left(-\frac{\epsilon_0\alpha}{2kT}\left|\frac{\partial\phi}{\partial y}\right|^2\right)-1\right]}\frac{\partial\phi}{\partial y}dy. \tag{3.7}$$

The steric-only equivalents of these expressions are Equations (2.13) and (2.14) respectively.

The projection of the Stokes equation on $\mathbf{e_x}$ can then be rewritten, for some y-coordinate value of y^*, as

$$\eta\frac{\partial^2 u}{\partial y^2} = -2zec_b\frac{\sinh\left(\frac{ze\phi}{kT}\right)\exp\left(-\frac{\epsilon_0\alpha}{2kT}\left|\frac{\partial\phi}{\partial y}\right|^2\right)}{1+\nu\left[\cosh\left(\frac{ze\phi}{kT}\right)\exp\left(-\frac{\epsilon_0\alpha}{2kT}\left|\frac{\partial\phi}{\partial y}\right|^2\right)-1\right]}\frac{\partial\psi}{\partial x}+$$
$$\frac{\partial}{\partial x}\int_\infty^{y^*} 2zec_b\frac{\sinh\left(\frac{ze\phi}{kT}\right)\exp\left(-\frac{\epsilon_0\alpha}{2kT}\left|\frac{\partial\phi}{\partial y}\right|^2\right)}{1+\nu\left[\cosh\left(\frac{ze\phi}{kT}\right)\exp\left(-\frac{\epsilon_0\alpha}{2kT}\left|\frac{\partial\phi}{\partial y}\right|^2\right)-1\right]}\frac{\partial\phi}{\partial y}dy. \quad (3.8)$$

This replaces its steric-only equivalent, Equation (2.16). Note that the coefficient of $\frac{\partial\psi}{\partial x}$ in the first term on the right hand side of Equation (3.8) does not reduce to $-\epsilon_m\frac{\partial^2\phi}{\partial y^2}$, but instead a linear combination of terms involving both $\frac{\partial^2\phi}{\partial y^2}$ and $\frac{\partial\phi}{\partial y}$. We can integrate Equation (3.8) twice with respect to y to obtain the following relation

$$\eta u_s = \int_0^\infty\int_{y''}^\infty 2zec_b\frac{\sinh\left(\frac{ze\phi}{kT}\right)\exp\left(-\frac{\epsilon_0\alpha}{2kT}\left|\frac{\partial\phi}{\partial y'}\right|^2\right)}{1+\nu\left[\cosh\left(\frac{ze\phi}{kT}\right)\exp\left(-\frac{\epsilon_0\alpha}{2kT}\left|\frac{\partial\phi}{\partial y'}\right|^2\right)-1\right]}\frac{\partial\psi}{\partial x}dy'dy''-$$
$$\int_0^\infty\int_{y''}^\infty\left(\frac{\partial}{\partial x}\int_\infty^{y'} 2zec_b\frac{\sinh\left(\frac{ze\phi}{kT}\right)\exp\left(-\frac{\epsilon_0\alpha}{2kT}\left|\frac{\partial\phi}{\partial y}\right|^2\right)}{1+\nu\left[\cosh\left(\frac{ze\phi}{kT}\right)\exp\left(-\frac{\epsilon_0\alpha}{2kT}\left|\frac{\partial\phi}{\partial y}\right|^2\right)-1\right]}\frac{\partial\phi}{\partial y}dy\right)dy'dy''. \quad (3.9)$$

Since the modified Poisson-Boltzmann relation is of a more complicated form in the dielectric decrement case than in the steric-only case, the integration of Equation (3.8) is not as straightforward as the integration of its steric-only equivalent. Next, keeping in mind that $\psi = \phi + \psi_b$, we can use Leibniz's rule and the first fundamental theorem

of calculus to modify Equation (3.9) to obtain

$$\eta u_s = \int_0^{\infty}\int_{y''}^{\infty}\int_{\infty}^{y'} 2zec_b\frac{\partial}{\partial\phi}\left(\frac{\sinh\left(\frac{ze\phi}{kT}\right)\exp\left(-\frac{\epsilon_0\alpha}{2kT}\left|\frac{\partial\phi}{\partial y}\right|^2\right)}{1+\nu\left[\cosh\left(\frac{ze\phi}{kT}\right)\exp\left(-\frac{\epsilon_0\alpha}{2kT}\left|\frac{\partial\phi}{\partial y}\right|^2\right)-1\right]}\frac{\partial\psi_b}{\partial x}\right)\frac{\partial\phi}{\partial y}dyd y' dy'' - \int_0^{\infty}\int_{y''}^{\infty}\int_{\infty}^{y'} 2ze\frac{\partial c_b}{\partial x}\frac{\sinh\left(\frac{ze\phi}{kT}\right)\exp\left(-\frac{\epsilon_0\alpha}{2kT}\left|\frac{\partial\phi}{\partial y}\right|^2\right)}{1+\nu\left[\cosh\left(\frac{ze\phi}{kT}\right)\exp\left(-\frac{\epsilon_0\alpha}{2kT}\left|\frac{\partial\phi}{\partial y}\right|^2\right)-1\right]}\frac{\partial\phi}{\partial y}dyd y' dy''. \quad (3.10)$$

In the case of $\alpha = 0$, the first term of the right hand side of Equation (3.10) reduces to a term proportional to the zeta potential as consistent with its steric-only equivalent, Equation (2.17). By completing the differentiation with respect to ϕ in the first term on the right hand side and performing changes of variables from y' to ϕ_1, from y'' to ϕ'', from ϕ_1 to $\phi'\phi''$ and from ϕ in the previous relation to $\phi\phi'\phi''$, we obtain the following expression for the slip velocity where the terms in the very large parentheses in the first two terms and the fraction following $\frac{\partial c_b}{\partial x}$ in the third term are dimensionless:

$$u_s = \frac{1}{\eta}\int_0^{\zeta}\int_0^1\int_0^1 \frac{\phi'\phi''^2}{\left(\frac{\partial(\phi'\phi'')}{\partial y}\right)\left(\frac{\partial\phi''}{\partial y}\right)}\Bigg[$$
$$\frac{2k^2T^2c_b}{ze}\frac{\partial\psi_b}{\partial x}\left(\frac{\cosh(\phi\phi'\phi'')\exp\left(-\alpha\left|\frac{\partial(\phi\phi'\phi'')}{\partial y}\right|^2\right)}{1+\nu\left[\cosh(\phi\phi'\phi'')\exp\left(-\alpha\left|\frac{\partial(\phi\phi'\phi'')}{\partial y}\right|^2\right)-1\right]}-\right.$$
$$\left.\frac{\nu\left[\sinh(\phi\phi'\phi'')\exp\left(-\alpha\left|\frac{\partial(\phi\phi'\phi'')}{\partial y}\right|^2\right)\right]^2}{\left(1+\nu\left[\cosh(\phi\phi'\phi'')\exp\left(-\alpha\left|\frac{\partial(\phi\phi'\phi'')}{\partial y}\right|^2\right)-1\right]\right)^2}\right)-$$
$$\frac{2k^2T^2\epsilon_w\epsilon_0\alpha}{z^2e^2}\frac{\partial(\phi\phi'\phi'')}{\partial y}\frac{\partial^2\psi_b}{\partial x\partial y}\left(\frac{\sinh(\phi\phi'\phi'')\exp\left(-\alpha\left|\frac{\partial(\phi\phi'\phi'')}{\partial y}\right|^2\right)}{1+\nu\left[\cosh(\phi\phi'\phi'')\exp\left(-\alpha\left|\frac{\partial(\phi\phi'\phi'')}{\partial y}\right|^2\right)-1\right]}-\right.$$
$$\left.\frac{\nu\sinh(\phi\phi'\phi'')\cosh(\phi\phi'\phi'')\left[\exp\left(-\alpha\left|\frac{\partial(\phi\phi'\phi'')}{\partial y}\right|^2\right)\right]^2}{\left(1+\nu\left[\cosh(\phi\phi'\phi'')\exp\left(-\alpha\left|\frac{\partial(\phi\phi'\phi'')}{\partial y}\right|^2\right)-1\right]\right)^2}\right)-$$
$$\left.\frac{2k^3T^3}{z^2e^2}\frac{\partial c_b}{\partial x}\frac{\sinh(\phi\phi'\phi'')\exp\left(-\alpha\left|\frac{\partial(\phi\phi'\phi'')}{\partial y}\right|^2\right)}{1+\nu\left[\cosh(\phi\phi'\phi'')\exp\left(-\alpha\left|\frac{\partial(\phi\phi'\phi'')}{\partial y}\right|^2\right)-1\right]}\right]d\phi d\phi' d\phi''. \quad (3.11)$$

In this expression, the second term of the right hand side goes to zero since ψ_b is a function of only x. Using the same scaling factor as in the steric-only model

$$U^* = \frac{\epsilon_m\varphi_T^2}{\eta R}. \quad (2.19)$$

to scale the velocity, and remembering that a factor of $\sqrt{c_b}$ is lost in the non-dimensionalization of the potential gradient, we can non-dimensionalize Equation (3.11)

to obtain

$$u_s = \int_0^{\zeta}\int_0^1\int_0^1 \frac{\phi'\phi''^2}{\left(\frac{\partial(\phi'\phi'')}{\partial y}\right)\left(\frac{\partial\phi''}{\partial y}\right)}\Bigg[\frac{\partial\psi_b}{\partial x}\left(\frac{\cosh(\phi\phi'\phi'')\exp\left(-\alpha\left|\frac{\partial(\phi\phi'\phi'')}{\partial y}\right|^2\right)}{1+\nu\left[\cosh(\phi\phi'\phi'')\exp\left(-\alpha\left|\frac{\partial(\phi\phi'\phi'')}{\partial y}\right|^2\right)-1\right]} - \frac{\nu\left[\sinh(\phi\phi'\phi'')\exp\left(-\alpha\left|\frac{\partial(\phi\phi'\phi'')}{\partial y}\right|^2\right)\right]^2}{\left(1+\nu\left[\cosh(\phi\phi'\phi'')\exp\left(-\alpha\left|\frac{\partial(\phi\phi'\phi'')}{\partial y}\right|^2\right)-1\right]\right)^2}\right) - \frac{1}{c_b}\frac{\partial c_b}{\partial x}\frac{\sinh(\phi\phi'\phi'')\exp\left(-\alpha\left|\frac{\partial(\phi\phi'\phi'')}{\partial y}\right|^2\right)}{1+\nu\left[\cosh(\phi\phi'\phi'')\exp\left(-\alpha\left|\frac{\partial(\phi\phi'\phi'')}{\partial y}\right|^2\right)-1\right]}\Bigg] d\phi d\phi' d\phi''. \quad (3.12)$$

3.1.3 Bulk solution: Excess charge and ion concentration

The analysis for the bulk solution remains almost identical to that in the case of the steric-only model, with the exception that the expressions for the excess charge q and the excess ion concentration w in the EDL need to be modified. The dimensional forms of q and w are given, in terms of dimensional α, by

$$q = -\int_0^{+\infty} 2zec_b \frac{\sinh\left(\frac{ze\phi}{kT}\right)\exp\left(-\frac{\epsilon_0\alpha}{2kT}\left|\frac{\partial\phi}{\partial y}\right|^2\right)}{1+\nu\left[\cosh\left(\frac{ze\phi}{kT}\right)\exp\left(-\frac{\epsilon_0\alpha}{2kT}\left|\frac{\partial\phi}{\partial y}\right|^2\right)-1\right]} dy \quad (3.13)$$

and

$$w = \int_0^{+\infty} 2c_b \left(\frac{\cosh\left(\frac{ze\phi}{kT}\right)\exp\left(-\frac{\epsilon_0\alpha}{2kT}\left|\frac{\partial\phi}{\partial y}\right|^2\right)}{1+\nu\left[\cosh\left(\frac{ze\phi}{kT}\right)\exp\left(-\frac{\epsilon_0\alpha}{2kT}\left|\frac{\partial\phi}{\partial y}\right|^2\right)-1\right]} - 1\right) dy, \quad (3.14)$$

while the dimensionless forms are given, in terms of dimensionless α, by

$$q = -\int_0^{+\infty} 2c_b \frac{\sinh(\phi)\exp\left(-\alpha\left|\frac{\partial\phi}{\partial y}\right|^2\right)}{1+\nu\left[\cosh(\phi)\exp\left(-\alpha\left|\frac{\partial\phi}{\partial y}\right|^2\right)-1\right]}dy \tag{3.15}$$

and

$$w = \int_0^{+\infty} 2c_b \left(\frac{\cosh(\phi)\exp\left(-\alpha\left|\frac{\partial\phi}{\partial y}\right|^2\right)}{1+\nu\left[\cosh(\phi)\exp\left(-\alpha\left|\frac{\partial\phi}{\partial y}\right|^2\right)-1\right]}-1\right)dy \tag{3.16}$$

using the same scaling constants q^* and w^* as in the steric-only model.

When advection is taken into consideration, the boundary conditions (2.35) in dimensional form and (2.57) in dimensionless form need to be adopted, with p_i given by the following modified expressions instead:

$$p_1 = \int_0^{+\infty} \Big[\{c_+(y)+c_-(y)\}\,\kappa_{D2}(\zeta,\phi(y)) - 2c_b\kappa_{D2}(\zeta,0)\Big]dy, \tag{3.17}$$

$$p_2 = \int_0^{+\infty} -\Big[\{c_+(y)+c_-(y)\}\,\kappa_{D1}(\zeta,\phi(y)) - 2c_b\kappa_{D1}(\zeta,0)\Big]dy, \tag{3.18}$$

$$p_3 = \int_0^{+\infty} \Big[\{c_+(y)-c_-(y)\}\,\kappa_{D2}(\zeta,\phi(y))\Big]dy, \tag{3.19}$$

$$p_4 = \int_0^{+\infty} -\Big[\{c_+(y)-c_-(y)\}\,\kappa_{D1}(\zeta,\phi(y))\Big]dy, \tag{3.20}$$

where $c_\pm$ are given by Equation (3.1), and $\kappa_{D1}(\zeta,\xi)$ and $\kappa_{D2}(\zeta,\xi)$ are given by

$$\kappa_{D1}(\zeta,\xi) = \int_\xi^\zeta \int_0^1 \int_0^1 \frac{\phi'\phi''^2}{\left(\frac{\partial(\phi'\phi'')}{\partial y}\right)\left(\frac{\partial\phi''}{\partial y}\right)} \Bigg(\frac{\cosh(\phi\phi'\phi'')\exp\left(-\alpha\left|\frac{\partial(\phi\phi'\phi'')}{\partial y}\right|^2\right)}{1+\nu\left[\cosh(\phi\phi'\phi'')\exp\left(-\alpha\left|\frac{\partial(\phi\phi'\phi'')}{\partial y}\right|^2\right)-1\right]} - \frac{\nu\left[\sinh(\phi\phi'\phi'')\exp\left(-\alpha\left|\frac{\partial(\phi\phi'\phi'')}{\partial y}\right|^2\right)\right]^2}{\left(1+\nu\left[\cosh(\phi\phi'\phi'')\exp\left(-\alpha\left|\frac{\partial(\phi\phi'\phi'')}{\partial y}\right|^2\right)-1\right]\right)^2} \Bigg) d\phi d\phi' d\phi'' \quad (3.21)$$

and

$$\kappa_{D2}(\zeta,\xi) = \int_\xi^\zeta \int_0^1 \int_0^1 \frac{\phi'\phi''^2}{\left(\frac{\partial(\phi'\phi'')}{\partial y}\right)\left(\frac{\partial\phi''}{\partial y}\right)} \Bigg(\frac{\sinh(\phi\phi'\phi'')\exp\left(-\alpha\left|\frac{\partial(\phi\phi'\phi'')}{\partial y}\right|^2\right)}{1+\nu\left[\cosh(\phi\phi'\phi'')\exp\left(-\alpha\left|\frac{\partial(\phi\phi'\phi'')}{\partial y}\right|^2\right)-1\right]} \Bigg) d\phi d\phi' d\phi''. \quad (3.22)$$

Observe that the expressions above are highly computationally intensive. Due to time constraints, we were unable to consider advection effects for the dielectric decrement model in our simulations for this work, but we present a possible implementation in Appendix A.6, which could potentially be simplified by evaluating fewer grid points and using interpolation for some of the functions.

3.2 Model simplifications

3.2.1 Simplified model without surface conduction

In the limit of zero surface conduction, which occurs in the limit of large particle radii or large bulk concentrations at infinity at low to moderate applied field strengths based on an analysis of Equation (2.57), concentration gradients around the particle are absent and the bulk concentration around the particle is everywhere uniform. In this scenario, the ionic concentration and electric potential fields are no longer coupled, and the electric potential is then a harmonic function that can be uniquely determined with the boundary conditions [39]

$$\Psi_b(r,\theta) = -E\left(r + \frac{1}{2r^2}\right)\cos\theta. \tag{3.23}$$

As a consequence, the zeta potential is

$$\zeta(\theta) = \Psi_P + \frac{3}{2}E\cos\theta. \tag{3.24}$$

It only remains, then, to determine the electric potential of the particle surface using the global conservation of charge in Equation (2.59), which can be re-expressed in dimensionless form as:

$$Q = -\frac{1}{2}\int_0^\pi q(\theta)\sin\theta d\theta, \tag{3.25}$$

where q is given by the relation (2.50) in the steric-only model and by the relation (3.15) in the model above including both steric and dielectric decrement effects. This simplified model will allow us to perform a quick comparison between the steric-only model and the dielectric decrement model including steric effects to determine the relative influence of the parameters ν and α.

3.2.2 Condensed layer approximation

This reduced model can be simplified even further at high E[1] to obtain an analytic expression for the excess charge q in the EDL and thus an asymptotic expression for the electrophoretic mobility μ_{EP}. We do so by assuming that the ionic concentration saturates near the particle surface. This is predicted by both the steric model and the dielectric decrement model: in the former, the finite size of ions prevents excessive accumulation of ions near the particle surface, and in the latter, the concentration dependence of ϵ_m prevents a excessively large ionic concentration near the particle surface since the dielectric permittivity cannot decrease too drastically. We can then divide the EDL into two regions: a region with saturated concentration near the particle surface ($y \to 0$), and an adjacent diffuse region where the ionic concentration decays to zero away from the particle surface according to the appropriately modified Poisson-Boltzmann relation. A similar partition of the EDL was performed by Kilic [4, 61], but with a less rigorous approximation for the diffuse region potential. (Note that q can be given by the expression $\int_0^\infty 2\rho_E \, dy$, where the factor of 2 appears due to the characteristic charge used in the non-dimensionalization process.)

Basic model

In the condensed layer, of approximate thickness L, we assume that the charge density due to the solvated ions reaches a maximum of ρ_{max}. The distribution of ions in the condensed layer is given approximately by the formula

$$\frac{\partial^2 \phi}{\partial y^2} = -\rho_{max}. \tag{3.26}$$

We can then write

$$\phi = -\frac{\rho_{max} y^2}{2} + Ay + \zeta \tag{3.27}$$

[1]Based on the discussion of the previous section, the high-field regime should not obey the limit of zero surface conduction except at exceedingly small ratios of Debye length to particle radius, or ϵ. We will assume this to be true to continue our analysis for the purpose of obtaining a better general understanding of electrophoretic phenomena at large fields, although we note that this reduced model has a much smaller region of physical validity than the one considered in the previous section.

for some constant A and zeta potential ζ, which is the value of ϕ at $y = 0$. (Recall that ϕ is the difference between the local electric potential in the double layer ψ and the potential at the bulk/double layer interface ψ_b.)

Using the Gouy-Chapman model, we can find the approximate distribution of ions in the adjoining diffuse layer. Assuming the potential has already decayed to a low value at the condensed layer/diffuse layer interface, this distribution is given approximately by the formula

$$\frac{\partial^2 \phi}{\partial y^2} = \sinh \phi = -\rho_E, \tag{3.28}$$

which in turn gives

$$\frac{\partial \phi}{\partial y} = -2 \sinh \frac{\phi}{2}. \tag{3.29}$$

Since the potential decays to zero at infinity by assumption, we can then write, using the general form of the solution given by Kirby [63],

$$\phi = 4 \tanh^{-1}(Be^{-y}) \tag{3.30}$$

for some constant B. This can also be written as

$$\tanh \frac{\phi}{4} = Be^{-y} \tag{3.31}$$

and as

$$\phi = 2 \ln \left(\frac{1 + Be^{-y}}{1 - Be^{-y}} \right), |Be^{-y}| < 1. \tag{3.32}$$

At $y = L$, we match the expressions for the potential and its space derivative in the condensed layer and the adjoining region to ensure the continuity and smoothness of the potential distribution, giving

$$\phi|_{y=L} = -\frac{\rho_{max} L^2}{2} + AL + \zeta = 4 \tanh^{-1}(Be^{-L}) \tag{3.33}$$

and

$$\left.\frac{\partial\phi}{\partial y}\right|_{y=L} = -\rho_{max}L + A = -2\sinh\frac{\phi|_{y=L}}{2}. \tag{3.34}$$

At $y = L$, we can also match the charge density expressions to give

$$\begin{aligned}\rho_E|_{y=L} = \rho_{max} &= -\sinh\phi|_{y=L} \\ &= -2\sinh\frac{\phi|_{y=L}}{2}\sqrt{1+\sinh^2\frac{\phi|_{y=L}}{2}},\end{aligned} \tag{3.35}$$

noting that the hyperbolic cosine function is always positive. Then, using Equations (3.34) and (3.35), we obtain

$$\rho_{max} = (A - \rho_{max}L)\sqrt{1+\left[\frac{A-\rho_{max}L}{-2}\right]^2}, \tag{3.36}$$

which simplifies to

$$4\rho_{max}^2 = 4(A-\rho_{max}L)^2 + (A-\rho_{max}L)^4. \tag{3.37}$$

Solving for the quadratic equation in $(A - \rho_{max}L)^2$ and taking the appropriate roots, we obtain

$$\begin{aligned}A &= \rho_{max}L \pm \sqrt{-2+2\sqrt{1+\rho_{max}^2}} \\ &= \rho_{max}L \pm c_1,\end{aligned} \tag{3.38}$$

where the constant c_1 is defined to simplify the subsequent derivation. (Observe, from Equation (3.34), that $\pm c_1 = -2\sinh\frac{\phi|_{y=L}}{2}$.)

Using the double angle formula for the hyperbolic sine function described in Equation (3.35) and the relation in Equation (3.31), we can solve for B, obtaining

$$\begin{aligned}B &= e^L\tanh\frac{\phi|_{y=L}}{4} \\ &= e^L\frac{c_2}{\sqrt{1+c_2^2}},\end{aligned} \tag{3.39}$$

where

$$c_2 = \pm\sqrt{-\frac{1}{2} + \frac{1}{2}\sqrt{\frac{1}{2} + \frac{1}{2}\sqrt{1+\rho_{max}^2}}}. \tag{3.40}$$

Finally, we can substitute the expression for A given in Equation (3.38) into Equation (3.33) and solve the resulting quadratic equation in L to obtain

$$L = \frac{\mp c_1 \pm \sqrt{c_1^2 - 2c_3\rho_{max}}}{\rho_{max}}, \tag{3.41}$$

where

$$\begin{aligned} c_3 &= \zeta - 2\ln\left(\frac{1+Be^{-L}}{1-Be^{-L}}\right) \\ &= \zeta - 2\ln\left(\frac{\sqrt{1+c_2^2}+c_2}{\sqrt{1+c_2^2}-c_2}\right). \end{aligned} \tag{3.42}$$

The total charge in the electric double layer is then given by

$$\begin{aligned} q &= 2\rho_{max}L + \int_L^\infty 2\rho_E \, dy \\ &= \pm 2\sqrt{c_1^2 - 2c_3\rho_{max}} \end{aligned} \tag{3.43}$$

Note that this expression is valid only when $c_1^2 > 2c_3\rho_{max}$. Since c_1^2 is always positive and ρ_{max} has a sign opposite to that of ζ, this condition will be violated if c_3 is of a sign opposite to that of ζ. This can occur if ζ is too small or if c_2 is too large. For the first condition, since we assumed earlier that we are operating in the high-field regime, the magnitudes of ψ_b and thus ζ can be taken to be large based on Equations (3.23) and (3.24) respectively. The first condition is thus not satisfied at large applied field strengths.[2] For the second condition, we will observe later that ρ_{max}, and thus c_2, increases with decreasing ν and α. The second condition is thus satisfied if both ν and α are too small. In this limit, ρ_{max} increases without bound, and it is true that the ionic concentration no longer saturates at the particle surface, but instead increases without bound. Then, a condensed layer model would no longer

[2]In addition, it is not essential to consider steric or dielectric decrement effects at small ζ.

be an accurate depiction of the physical situation. Hence, a successful application of the condensed layer model requires sufficiently large values of E, ν and α.

Note, also, that q is a function of ρ_{max}, which is constant for a given physical configuration, and ζ only. We can thus express q in the form

$$q = \pm c_6 \sqrt{c_4 + c_5 \zeta} \tag{3.44}$$

where

$$c_4 = c_1^2 + 4\rho_{max} \ln \left(\frac{\sqrt{1 + c_2^2} + c_2}{\sqrt{1 + c_2^2} - c_2} \right), \tag{3.45}$$

$$c_5 = -2\rho_{max}, \tag{3.46}$$

and $c_6 = 2$. The dimensionless charge conservation relation (3.25) can then be expressed as

$$\begin{aligned} Q &= \frac{c_6}{3E} \int_{\zeta(\pi)}^{\zeta(0)} \sqrt{c_4 + c_5 \zeta} d\zeta \\ &= \frac{c_6}{3E} \int_{-\frac{3}{2}E + \Psi_P}^{\frac{3}{2}E + \Psi_P} \sqrt{c_4 + c_5 \zeta} d\zeta \\ &= \frac{c_6}{3c_5 E} \int_{\frac{3}{2}E - \Psi_P}^{\frac{3}{2}E + \Psi_P} c_5 \sqrt{c_4 + c_5 \zeta} d\zeta \\ &= \frac{2c_6}{9c_5 E} \left[(c_4 + c_5 \zeta)^{3/2} \right]_{\frac{3}{2}E - \Psi_P}^{\frac{3}{2}E + \Psi_P} \\ &\approx \frac{2c_6}{3E} \Psi_P \sqrt{c_4 + \frac{3}{2} c_5 E}. \end{aligned} \tag{3.47}$$

The modification of the lower limit of the integral is permissible since the integrand is uneven with respect to ζ, and the final result is obtained with a binomial approximation assuming $\frac{3}{2}E \gg \Psi_P$, which is true for high field strengths. We then obtain, after substituting c_6,

$$\Psi_P = \frac{3QE}{4} \frac{1}{\sqrt{c_4 + \frac{3}{2} c_5 E}}, \tag{3.48}$$

and finally an expression for the mobility

$$\mu_{EP} = \frac{3}{2}\Psi_P = \frac{9QE}{8}\frac{1}{\sqrt{c_4 + \frac{3}{2}c_5 E}}. \tag{3.49}$$

The factor of $\frac{3}{2}$ arises by observing that the slip velocity scales as $\zeta\frac{\partial\Psi_b}{\partial\theta}$, and then averaging this value for the slip velocity over all angles and scaling by the electric field to obtain a value for the mobility [39].

Adding dielectric decrement effects

The basic model presented above uses the standard form of the Poisson-Boltzmann relation equating the second space derivative of the potential to the negative of the charge density, and assumes a constant dielectric permittivity. From the analysis earlier in this Chapter, we know that this is not an accurate depiction of the effects of dielectric decrement. We attempt to resolve this by assuming a reduced dielectric permittivity in the condensed layer[3]

$$\epsilon_{DD} = \epsilon_{bulk}\left(1 - 2\alpha\cosh\phi|_{y=L}\right) = c_7\epsilon_{bulk}, \tag{3.50}$$

using an expression for the sum of the cationic and anionic concentrations consistent with $\rho_{max} = -\sinh\phi|_{y=L}$. Equation (3.26) can then be written as

$$\frac{\partial^2\phi}{\partial y^2} = -\frac{1}{c_7}\rho_{max}. \tag{3.51}$$

The constant A will then have to be modified to the following expression

$$A = \frac{1}{c_7}\rho_{max}L \pm c_1, \tag{3.52}$$

[3]Note that this causes the dielectric permittivity to be discontinuous at the condensed layer/diffuse layer interface.

while c_1, c_2, c_3 and B all remain the same. The expression for L will also have to be rewritten as

$$L = c_7 \frac{\mp c_1 \pm \sqrt{c_1^2 - \dfrac{2c_3\rho_{max}}{c_7}}}{\rho_{max}}, \tag{3.53}$$

giving a modified expression for q

$$\begin{aligned} q &= \pm 2(1-c_7)c_1 \pm 2c_7\sqrt{c_1^2 - \frac{2c_3\rho_{max}}{c_7}} \\ &= \pm 2(1-c_7)c_1 \pm 2c_7\sqrt{c_4' + c_5'\zeta}, \end{aligned} \tag{3.54}$$

where

$$c_4' = c_1^2 + 4\frac{\rho_{max}}{c_7}\ln\left(\frac{\sqrt{1+c_2^2}+c_2}{\sqrt{1+c_2^2}-c_2}\right) \tag{3.55}$$

and

$$c_5' = -2\frac{\rho_{max}}{c_7}. \tag{3.56}$$

This gives us the following modified expressions

$$Q \approx \frac{4c_7}{3E}\Psi_P\sqrt{c_4' + \frac{3}{2}c_5'E} + \frac{8c_1c_7(1-c_7)}{3E}\Psi_P \tag{3.57}$$

and

$$\mu_{EP} = \frac{9QE}{8c_7}\frac{1}{\sqrt{c_4' + \dfrac{3}{2}c_5'E} + 2c_1(1-c_7)}. \tag{3.58}$$

Saturation concentrations

We now derive the values of ρ_{max} for the various models.

For the steric model, we have

$$\rho_{max} = \lim_{|\phi|\to\infty}\frac{-c_b\sinh\phi}{1+\nu(\cosh\phi - 1)} = -\text{sign}(\phi)\frac{c_b}{\nu}. \tag{3.59}$$

This can be seen by noting that in the limit of high ϕ, $\cosh\phi$ can be approximated as $e^{\phi}/2$ if $\phi > 0$ or $e^{-\phi}/2$ if $\phi < 0$, while $\sinh\phi$ can be approximated as $e^{\phi}/2$ if $\phi > 0$, or $-e^{-\phi}/2$ if $\phi < 0$. Then, approximating the denominator in the expression above

as $\cosh\phi$, we obtain the desired final result for ρ_{max}.

For the dielectric decrement model, when 4α is sufficiently larger than ν, we obtain the following limit [59]

$$\rho_{max} = \lim_{|\phi|\to\infty} \frac{-c_b \sinh\phi \exp(-\alpha\phi'^2)}{1+\nu[\cosh\phi\exp(-\alpha\phi'^2)-1]} = -\text{sign}(\phi)\frac{c_b}{4\alpha}. \tag{3.60}$$

The derivation of this expression is more involved. Let us consider the derivation of the dielectric decrement equations neglecting steric effects. From the Gibbs-Duhem relation for a two component system at constant temperature, we know that

$$-Vd\Pi + N_+ d\mu_+ + N_- d\mu_- = 0 \tag{3.61}$$

where V is the volume of the system under consideration, Π is the osmotic pressure of the system, $N_\pm$ is the number of cations/anions in the system and $\mu_\pm$ is the cationic/anionic chemical potential. From the derivation of Equation (3.1), and by noting that the entropic contribution to the Helmholtz free energy involves Stirling's approximation [61], we can deduce that the chemical potential of the ions is

$$\mu_\pm = \frac{\epsilon_0}{2}\alpha\left|\nabla\phi\right|^2 \pm ze\phi + kT\ln(c_\pm). \tag{3.62}$$

We note that $\mu_\pm = \mu_\pm(c_\pm, \phi, \nabla\phi)$, $\phi = \phi(c_\pm, y)$, $c_\pm = N_\pm/V$, and $\left[\frac{\partial c}{\partial y}\right]_\phi \left[\frac{\partial y}{\partial \phi}\right]_c \left[\frac{\partial \phi}{\partial c}\right]_y =$

-1. This allows us to rewrite the Gibbs-Duhem relation as

$$\begin{aligned}
d\Pi &= \epsilon_0\alpha(c_+ + c_-)|\nabla\phi|d(|\nabla\phi|) + ze(c_+ - c_-)d\phi + kT(dc_+ + dc_-) \\
&= \epsilon_0\alpha(c_+ + c_-)|\nabla\phi|d(|\nabla\phi|) - \nabla\cdot(\epsilon_m\nabla\phi)d\phi + kT(dc_+ + dc_-) \\
&= \epsilon_0\alpha(c_+ + c_-)|\nabla\phi|d(|\nabla\phi|) - \epsilon_m\nabla^2\phi d\phi - \nabla\epsilon_m\nabla\phi d\phi + kT(dc_+ + dc_-) \\
&= \epsilon_0\alpha(c_+ + c_-)|\nabla\phi|d(|\nabla\phi|) - \epsilon_m\nabla^2\phi d\phi \\
&\quad - \nabla\epsilon_m\nabla\phi\left(\frac{\partial\phi}{\partial y}dy + \frac{\partial\phi}{\partial c_+}dc_+ + \frac{\partial\phi}{\partial c_-}dc_-\right) + kT(dc_+ + dc_-) \\
&= \epsilon_0\alpha(c_+ + c_-)|\nabla\phi|d(|\nabla\phi|) - \frac{1}{2}\epsilon_m d\left([\nabla\phi]^2\right) - (-\alpha\epsilon_0)|\nabla\phi|^2(dc_+ + dc_-) \\
&\quad - (-\alpha\epsilon_0)\left(\nabla c_+\frac{\partial\phi}{\partial c_+}dc_+ + \nabla c_-\frac{\partial\phi}{\partial c_-}dc_-\right)\nabla\phi + kT(dc_+ + dc_-) \\
&= \epsilon_0\alpha(c_+ + c_-)|\nabla\phi|d(|\nabla\phi|) - \frac{1}{2}\epsilon_m d\left([\nabla\phi]^2\right) - (-\alpha\epsilon_0)|\nabla\phi|^2(dc_+ + dc_-) \\
&\quad + (-\alpha\epsilon_0)|\nabla\phi|^2(dc_+ + dc_-) + kT(dc_+ + dc_-) \\
&= \epsilon_0\alpha(c_+ + c_-)|\nabla\phi|d(|\nabla\phi|) - \frac{1}{2}\epsilon_m d\left([\nabla\phi]^2\right) + kT(dc_+ + dc_-), \qquad (3.63)
\end{aligned}$$

By integrating this expression from an infinite system volume to the desired system volume, we obtain

$$\Pi = \frac{1}{2}\epsilon_0\alpha(c_+ + c_-)|\nabla\phi|^2 - \frac{1}{2}\epsilon_0(\epsilon_w - \alpha(c_+ + c_-))|\nabla\phi|^2 + kT(c_+ + c_- - 2). \quad (3.64)$$

Then, by non-dimensionalizing the various quantities and reducing the equation to its one-dimensional form, we obtain the expression

$$\Pi = -\frac{1}{2}\left([1 - \alpha(c_+ + c_-)] - \alpha(c_+ + c_-)\right)\left(\frac{\partial\phi}{\partial y}\right)^2 + (c_+ + c_- - 2) \qquad (3.65)$$

where Π is an invariant of the system. The same expression was derived by Ben-Yaakov *et al.* [56] (with the difference of an integration constant). This leads us to the result

$$\left(1 - 4\alpha\cosh(\phi)\exp(-\alpha\phi'^2)\right)\phi'^2 = 4\cosh(\phi)\exp(-\alpha\phi'^2) - 4 \qquad (3.66)$$

and thus

$$\cosh(\phi)\exp(-\alpha\phi'^2) = \frac{4+\phi'^2}{4+4\alpha\phi'^2}. \tag{3.67}$$

By taking the limit $|\phi| \to \infty$ and thus $|\phi'| \to \infty$, we obtain $\cosh(\phi)\exp(-\alpha\phi'^2) \to 1/4\alpha$ and $\sinh(\phi)\exp(-\alpha\phi'^2) \to \text{sign}(\phi)\cosh(\phi)\exp(-\alpha\phi'^2)$, and thus the desired result for ρ_{max}. A similar result was derived by Zhao and Zhai [59].

At smaller α, we obtain the following limit

$$\rho_{max} = \lim_{|\phi|\to\infty} \frac{-c_b \sinh\phi \exp(-\alpha\phi'^2)}{1+\nu[\cosh\phi\exp(-\alpha\phi'^2)-1]} = -\text{sign}(\phi)\frac{c_b}{4\alpha+\nu(1-4\alpha)}. \tag{3.68}$$

We will observe later that only one of the two quantities α and ν tends to dominate the physical structure of the EDL: if the values of ρ_{max} given by Equations (3.59) and (3.60) are dissimilar, the concentration tends to saturate at the smaller of the two values. Hence, the limit in Equation (3.68) should only be used if $\nu \approx 4\alpha$.

Refinement for steric-only model

In the steric-only model, the analytic expression (2.50) for q allows us an alternative approximation for the electrophoretic mobility. At large E and reasonably large ν, we can make the following approximation

$$\sqrt{\frac{2}{\nu}\ln\left[1+2\nu\sinh^2\left(\frac{\zeta}{2}\right)\right]} \approx \sqrt{\frac{2}{\nu}\left(\ln\frac{\nu}{2}+\zeta\right)}. \tag{3.69}$$

Based on the derivation of Equation (3.47), this gives us

$$\mu_{EP} = \frac{9QE}{8}\sqrt{\frac{\nu}{2\ln\frac{\nu}{2}+3E}}. \tag{3.70}$$

At very large values of E, the limit of the relation becomes

$$\mu_{EP} = \frac{3Q\sqrt{3\nu E}}{8} \propto \sqrt{E}, \tag{3.71}$$

a result which was first found and interpreted by Bazant *et al.* [6] in 2009.

Chapter 4

Charge-induced thickening

4.1 Motivation

This study on charge-induced thickening is primarily motivated by postulates in the electrophoretic literature that discrepancies in the prediction of electrophoretic mobilities could be caused by an underestimate of the viscosity in the electric double layers surrounding the electrophoretic particles [6, 64]. In particular, Bazant *et al.* [6] notes that this charge-induced thickening could be caused by the physical jamming and alignment of ions near the particle surface, and Coulombic interactions between the ions. In their review paper, they postulate the following arbitrary relation

$$\frac{\eta}{\eta_b} = \frac{1}{1 - \frac{\phi}{\phi_m}}, \tag{4.1}$$

which gives the local solvent viscosity η at some point in the double layer with a local ionic volume fraction of ϕ.[1] Here, η_b is the bulk solvent viscosity with no additional solvated ions, and ϕ_m is the maximum physically possible volume fraction that ions can take up in the solvent. A study of the rheological literature has revealed other validated empirical relations, as well as relations derived from more fundamen-

[1]We recognize that the symbol ϕ was used earlier to denote electric potential. However, ϕ is a common symbol for volume fraction in the rheological literature, and thus we will continue to use ϕ as a measure of volume fraction in this Chapter. We will express this fraction in terms of ν later to avoid its uage, and will alert the reader once we end the usage of ϕ as a measure of volume fraction.

tal physical principles. In this study, we offer an overview of this literature, and highlight some of these relations that could possibly be extended to our investigation of electrophoresis.

4.2 Preliminary assumptions

The bulk of the theoretical rheological literature deals with the increase in viscosity of a solvent with a significant volume fraction of solvated particles. This involves the fundamental assumption that the solvent can be treated as a continuum, which in turn requires that the size of the particles be significantly larger than the size of the solvent molecules. In their study of microrheology, several authors [65, 66] emphasized the importance for this continuum approximation to be satisfied even as one starts to deal with smaller particles. While we recognize that the size of a typical ion is much smaller than the sizes of the particles dealt with in the rheological literature and still somewhat smaller than those in the papers dealing with microrheology, we will continue to retain this assumption for the sake of preliminary modeling. In particular, most ions in aqueous solution are surrounded by solvation shells. We note that a more accurate model may involve a more fundamental approach that deals with the interactions between ions and solvent molecules, although in this discrete limit, the concept of a dynamic viscosity for the solvent begins to lose its meaning as well.

The viscosity of the fluid depends on electric and diffusive effects as well, and these in turn depend on the external electric field strength and the concentration gradients present in the fluid respectively. For the moment, we assume that shear viscous effects can be isolated in order to determine the amount by which to modify the viscosity to account for ion packing effects, and that the electric and diffusive effects can later be accounted for using the Navier-Stokes and Poisson-Nernst-Planck equations used in our model. As such, we assume that the ionic volume fraction in the immediate vicinity of an ion is uniform and that the rheological properties of the solvation shells - the analog of double layers around larger particles - are not affected

by the electric field. This has in fact been found to be an inaccurate assumption: in particular, Lyklema [67] notes that the "primary electroviscous effect" caused by the hydrodynamical interaction between the solvation shells around ions and the remainder of the surrounding solvent can be significant, especially for small ions. However, we will first retain this assumption, again for preliminary modeling purposes, and then discuss its validity later as we discuss the existing empirical relations for ions.

Last but not least, we will see later that it is important to determine if the flow at any point is a pure straining motion or one that involves shear. We assume that the flow in the double layer region surrounding the particle is a high shear region given the large variation of electroosmotic speed over a small double layer thickness, and that the flow in the bulk can be approximated as a pure strain flow with negligible shear.

4.3 Empirical relations for particles

Many empirical fits have been obtained for the increase in viscosity due to an increase in the volume fraction of solvated particles and outlined by several authors [68, 69, 70]. These fits involve both Newtonian and non-Newtonian relations. We will offer a non-exhaustive list of some of the Newtonian fits here.

The most well-known empirical fit appears to be that of Krieger and Dougherty [71], who obtained the following fit

$$\frac{\eta}{\eta_b} = \left(\frac{1}{1 - \frac{\phi}{\phi_m}} \right)^{2.5\phi_m}, \tag{4.2}$$

based on a modification of Mooney's fit [72]

$$\ln\left(\frac{\eta}{\eta_b}\right) = \frac{2.5\phi}{1 - \frac{\phi}{\phi_m}}. \tag{4.3}$$

Other relations include the fit obtained by Eilers [73]:

$$\frac{\eta}{\eta_b} = \left(1 + \frac{1.25\phi}{1 - \frac{\phi}{\phi_m}}\right)^2, \tag{4.4}$$

the fit obtained by Chong *et al.* [74]:

$$\frac{\eta}{\eta_b} = \left(1 + \frac{0.75\frac{\phi}{\phi_m}}{1 - \frac{\phi}{\phi_m}}\right)^2, \tag{4.5}$$

the fit obtained by Chang and Powell [75]:

$$\frac{\eta}{\eta_b} = \left(\frac{1}{1 - 1.033\frac{\phi}{\phi_m}}\right)^{1.8}, \tag{4.6}$$

and the fit obtained by Probstein and his collaborators [76, 77], which is given approximately by:

$$\frac{\eta}{\eta_b} = 1 + \frac{4.5\pi}{8}\frac{\beta}{\beta+1}\left(\frac{3 + 4.5\beta + \beta^2}{\beta+1} - 3\left(1 + \frac{1}{\beta}\right)\ln(\beta+1)\right) \tag{4.7}$$

where

$$\beta = \frac{\left(\frac{\phi}{\phi_m}\right)^{1/3}}{1 - \left(\frac{\phi}{\phi_m}\right)^{1/3}}. \tag{4.8}$$

The last fit is only a good fit for suspensions with high $\phi > 0.2$.

It has been noted that the choice of ϕ_m remains relatively arbitrary despite much study [68]. A range of values for ϕ_m has been used for the fits above, from the simple cubic value of 0.524 to the hexagonal close-packed value of 0.740. It has been conventional recently to adopt a random close-packed value of approximately 0.63, even though it has also been recognised that ϕ_m could vary with the constituents and the initial configuration of the system itself.

4.4 Einstein's theoretical relation and its extensions

Having examined several empirical relations that tie viscosity and volume fraction together, we now turn to a theoretical approach to obtain more physical insights into this phenomenon. Einstein [78] analyzed the perturbation of the velocity field of a fluid with irrotational Stokes flow due to the addition of a particle to it, as well as the work done on the fluid due to the interaction between the particle and the surrounding fluid. By equating this work with the energy dissipated due to viscous stresses and ensuring that the principle of continuity remained valid throughout the solvent, Einstein was able to obtain the following relation for dilute (i.e. low ϕ) and uniform suspensions of force-free particles

$$\frac{\eta}{\eta_b} = 1 + 2.5\phi. \tag{4.9}$$

Batchelor and Green [79, 80, 81, 82] extended Einstein's work for dilute suspensions by considering hydrodynamic interactions between particles, as well as Brownian motion, obtaining the following relation when Brownian motion is significant

$$\frac{\eta}{\eta_b} = 1 + 2.5\phi + 6.2\phi^2 \tag{4.10}$$

where the coefficient of ϕ^2 is made up of two contributions: a contribution of 5.2 (later revised to 5.0 by other authors [83] based on more accurate numerical computations) from hydrodynamic interactions using a uniform pair distribution function for the particles, and a contribution of 1.0 from Brownian motion. Batchelor also obtained a coefficient of 7.6 for the ϕ^2 term for non-Brownian suspensions (where Brownian motion is no longer significant) where shear flow is not present, comprising the same contribution of 5.2 from hydrodynamic interactions and 2.4 from a non-uniform (but isotropic) pair distribution function. He was unable to obtain a coefficient for non-Brownian suspensions with shear flow, citing difficulties in identifying a suitable pair

distribution function. It has been noted that both Einstein's and Batchelor's relations underestimate the viscosity of real suspensions [68].

Using a method of reflections instead of a specific pair distribution function, other authors have independently obtained other coefficients for the ϕ^2 term. Guth and Simha [84] obtained the following equation (with errors in the original derivation corrected)

$$\frac{\eta}{\eta_b} = 1 + 2.5\phi + 14.1\phi^2, \tag{4.11}$$

while Vand [85] obtained the following equation

$$\frac{\eta}{\eta_b} = 1 + 2.5\phi + 7.349\phi^2. \tag{4.12}$$

Simha [86] later refined the ϕ^2 coefficient to 12.6, citing excluded volume effects. Also, by solving the potential field of a suspension of fluid particles using the electrical potential analog problem, Kynch [87, 88] obtained the following equation

$$\frac{\eta}{\eta_b} = 1 + 2.5\phi + 7.5\phi^2. \tag{4.13}$$

Russel and his collaborators [89, 90, 91] extended Batchelor's work by accounting for thermodynamic effects in the pair distribution function for steady (non-zero) shear flow in the low-shear limit without neglecting Brownian motion, thus obtaining a coefficient of ϕ^2 that generally increased with ϕ. While they were able to obtain higher viscosities at higher volume fractions than those predicted by Batchelor, they were still unable to obtain a relation that closely matched experimental data at these high volume fractions. They postulated that this could have been because their assumption that hydrodynamic interactions were pairwise additive was probably invalid. A similar trend was observed in the case of zero shear flow (i.e. the high-frequency limit of shear flow) as well, although the authors did not specify the ϕ^2 coefficients obtained in this case.

Many authors extended Batchelor's approach in more recent years by including three-body interactions in their derivations. Cichocki *et al.* [83] obtained the following

equation (correcting Batchelor's coefficient for ϕ^2) for zero shear flow

$$\frac{\eta}{\eta_b} = 1 + 2.5\phi + 5.0023\phi^2 + 9.09\phi^3, \tag{4.14}$$

while Thomas and Muthukumar [92] obtained the following equation for steady shear in the low-shear limit in an earlier paper

$$\frac{\eta}{\eta_b} = 1 + 2.5\phi + 4.68\phi^2 + 6.40\phi^3. \tag{4.15}$$

Thomas and Muthukumar also offer virial expansions (up to the ϕ^3 term) of the Krieger-Dougherty equation

$$\frac{\eta}{\eta_b} = 1 + 2.5\phi + 4.81\phi^2 + 8.35\phi^3 \tag{4.16}$$

and the Mooney equation

$$\frac{\eta}{\eta_b} = 1 + 2.5\phi + 5.00\phi^2 + 8.69\phi^3 \tag{4.17}$$

for comparison, using a ϕ_m value of about 0.75. (Note that these empirical relations can be fitted to both zero shear and steady shear flows simply with an appropriate choice of ϕ_m. Also, Batchelor's coefficient for ϕ^2 seems to match the empirical fits more closely than those of the other papers cited earlier.)

Some authors [93, 83] have noted that the empirical fits appear to be matched rather closely by the results of Beenakker [94]. Beenakker used a wave vector approach to determine the relative viscosity, and noted that this is equivalent to taking into account many-body hydrodynamic interactions. Because the relative viscosity is directly derived from the wave vector dependence, no direct relation between viscosity and volume fraction was offered by Beenakker in his paper, but we were able to obtain the following cubic fit for Beenaker's relative viscosities in the zero shear case:

$$\frac{\eta}{\eta_b} = 1 + 2.5\phi + 5.2\phi^2 + 7.4\phi^3. \tag{4.18}$$

Beenakker predicts higher relative viscosities in the steady shear case.

Before we conclude this section, let us return to the assumption that electric effects can be neglected in the analysis. Several authors [95, 96, 97] have shown that Einstein's ϕ coefficient needs to be modified to account for the interaction between the double layer around a particle and the fluid it is immersed in. Lyklema [67] generalizes this further, noting that Einstein's ϕ coefficient needs to be corrected for the primary electroviscous effect stemming from "the energy dissipation caused by distortion of the double layer by shear", while the coefficient of ϕ^2 needs to be corrected for the secondary electroviscous effect coming from "the influence of pair interaction, electric repulsion particularly on the viscosity". We will thus examine this assumption further in the subsequent section.

4.5 Empirical relations for ions

Having illustrated the theoretical approach for determining the variation of relative viscosity with the volume fraction of suspended particles, we are now in a position to discuss existing empirical relations for the variation of relative viscosity with the ionic concentration in electrolytes.

In 1929, Jones and Dole [98] obtained the following fit for the change in viscosity resulting from a change in the ionic concentration c of the electrolyte (re-casted here for convenience):

$$\frac{\eta}{\eta_b} = 1 + A\sqrt{c} + Bc \tag{4.19}$$

for some constants A and B. It was noted that the value of A is positive for strong electrolytes and zero for non-electrolytes, suggesting an influence from inter-ion interactions, and the value of B could be either positive or negative. Both constants were temperature-dependent. The Jones-Dole equation was extended by Kaminsky [99] to the following relation

$$\frac{\eta}{\eta_b} = 1 + A\sqrt{c} + Bc + Dc^2 \tag{4.20}$$

to fit experiments dealing with higher concentrations. It has been noted [67] that

the term involving A is much less than unity, and thus can be neglected when the volume fraction becomes significant. Hence, we will ignore this term in the subsequent discussion.

Attempts have been made to reconcile the Jones-Dole equation (or the Kaminsky extension) with the correlation obtained by Einstein (or Batchelor's extension). Some authors [100, 101] were able to obtain a reasonable fit between the two by assuming appropriate effective ionic volumes, while other authors [102, 103, 104, 105, 106, 107] were able to obtain a reasonable fit only for some ions. It was found that Einstein's relation underestimates the B coefficient for smaller cations [106, 107], and some authors expressed skepticism in applying Einstein's relation to ions using structure-free ionic volumes [103, 108], but other authors [104, 105, 109] were able to account for this by including the volume of the solvation shell in determining the effective ionic volume of these cations. It was also determined that Einstein's relation could be modified [106, 107] to account for the volume of the solvent particles in order to accommodate the negative B coefficients, but this results in a prediction of excessive solvation of anions that are not known to be surrounded by a solvation shell [104, 105], although a modified model of solvation could possibly resolve this. Nevertheless, Einstein's relation cannot be modified to accommodate the temperature dependence of B [108].

Marcus [110] noted that a theoretical understanding of the Jones-Dole equation remained inadequate, but that a comprehensive list of experimentally-derived B coefficients was available in the literature [111]. The D coefficient is even less well-understood, and uncertainties in the additivity of the cationic and anionic coefficients [103] and the dependence of the coefficient on temperature [108] remain. In addition, this author is not aware of any compilation of experimental data for D coefficients at the moment due to the relative lack of data. A recent attempt [109] was made to rationalize the B coefficients using the limiting Gibbs energy of activation, but this approach remains partially empirical in nature and has not been successfully extended to the D coefficients. Knowledge of the D coefficients is probably necessary in our case to account for the high ionic concentrations in the double layer, and a

lack of experimental data in this area could compromise the utility of the Kaminsky equation.

4.6 Theoretical and numerical relations for ions

A approach similar to that adopted by Beenakker for particles was adopted by Chandra and Bagchi [112] to provide a theoretical background for the term involving A in the Jones-Dole equation. This work was later extended to demonstrate that shear relaxation effects (the primary electroviscous effect) and ion-ion interactions (the secondary electroviscous effect) were only dominant at low concentrations ($\phi < 0.05$)[2] where correction to the relative viscosity is necessary but only involving the term containing A in the Jones-Dole equation [113]. Thus, it appears that a general relation based on Einstein's and Batchelor's analysis for larger particles could serve as a reasonably accurate first approximation for our purposes, although it must be recognized that applying their analysis to ions has its inherent limitations.

It should be noted that other theoretical models have been derived [114, 115] independently of the Jones-Dole equation to account for the viscosity of electrolyte solutions, but they remain insufficiently general for our purposes.

4.7 Saitô's theoretical relation and its extensions

Using the invariance of shear forces over the perturbation of the velocity field resulting from the addition of a particle to a fluid, Saitô [116, 117] obtained the following relation

$$\frac{\eta}{\eta_b} = \frac{2.5\phi}{1-\phi}, \tag{4.21}$$

which can be re-written as

$$\frac{\frac{\eta}{\eta_b} - 1}{\frac{\eta}{\eta_b} + 1.5} = \phi. \tag{4.22}$$

[2]While we will demonstrate later that values for ν are typically on the order of 1.0×10^{-5} to 0.1, this is only an expression of steric packing in the solvent as a whole; in the condensed layer of ions close to the particle surface, ϕ will reach much higher values due to the accumulation of ions.

Saitô's equation is remarkably a good fit to experimental data [70] for an expression derived from first principles, although Beenakker [94] notes that the relation fails to account for correlations between particles.

Bedeaux [118] modified this relation to handle particle-particle and particle-solvent interactions after comparing it to the Clausius-Mosotti relation for the dielectric constant, thereby obtaining

$$\frac{\frac{\eta}{\eta_b} - 1}{\frac{\eta}{\eta_b} + 1.5} = \phi \left[1 + S(\phi)\right] \tag{4.23}$$

for some function $S(\phi)$.

van der Werff [119] obtained the following expression in the zero-shear limit

$$S(\phi) = 1.41\phi - 1.19\phi^2 \tag{4.24}$$

after a fit to experimental data (and again remarked the good fit between Beenakker's results and the experimental data), while Ladd [120] obtained the following expression

$$S(\phi) = \phi + \phi^2 - 2.3\phi^3 \tag{4.25}$$

after a fit to data from numerical simulations, noting that deviations from van der Werff's results are minimal for $\phi < 0.35$. Ladd also remarked that his expression for $S(\phi)$ offered a better fit to some experimental data than van der Werff's fit. Cichocki [83] also obtained an expression of

$$S(\phi) = 1.0009\phi + 0.63\phi^2 \tag{4.26}$$

after a fit to data from numerical simulations.

4.8 Synthesis

For the zero-shear limit, which the bulk solution in our model obeys, two equations in the above discussion appear to be good first approximations to model the phenomenon

of charge-induced thickening in a manner that can be applied to most ions. These are the empirical Krieger-Dougherty equation

$$\frac{\eta}{\eta_b} = \left(\frac{1}{1 - \frac{\phi}{\phi_m}} \right)^{2.5\phi_m}, \tag{4.2}$$

which is supported by Beenakker's theoretical analysis, and the generalized Saitô-Bedeaux equation

$$\frac{\frac{\eta}{\eta_b} - 1}{\frac{\eta}{\eta_b} + 1.5} = \phi \left[1 + S(\phi) \right] \tag{4.23}$$

using Ladd's formula

$$S(\phi) = \phi + \phi^2 - 2.3\phi^3, \tag{4.25}$$

which is also supported by Beenakker's data.

The high-shear limit is a trickier situation, but we claim that the same two relations above can also be used as reasonable first approximations. In a paper discussing the sedimentation of suspended spheres, Batchelor [121] notes that in the limit of high shear and thus high Péclet numbers, a suspension of neutrally buoyant spheres of approximately the same size in the zero-diffusion limit has a uniform pair distribution function. This means that the expressions for the zero-shear limit can be used in the high-shear limit as well. Bergenholtz *et al.* [122] also remarked, following numerical simulations, that at high Péclet numbers and exceedingly small particle separations, only hydrodynamic interactions are significant and the flow becomes Newtonian, provided interparticle attractions are not strong enough to result in shear thickening. In particular, it is demonstrated that the relative viscosity in the high-shear limit approaches the relative viscosity in the zero-shear limit. These conditions are obeyed in the double layer in our model, which comprises closely-packed counterions with the same electric charge.

4.9 Modification of numerical model

At this point, it is convenient to express the local volume fraction ϕ in terms of the local ionic concentration[3] c and the parameter ν used for steric packing earlier

$$\begin{aligned}\phi &= \frac{V_{ion}}{V_{total}} \\ &= \frac{4}{3}\pi\left(\frac{a}{2}\right)^3 c \\ &= \frac{c\nu\pi}{12c_{b,\infty}}.\end{aligned} \tag{4.27}$$

From here on, we will stop using ϕ to denote the ionic volume fraction, and return to its original usage as a symbol for electric potential.

For simplicity, we will apply the charge-induced thickening model to the steric-only model described in Chapter 2. Charge-induced thickening effects can be completely modeled by modifying the Navier-Stokes equations in both the EDL and the bulk solution. In the EDL, we replace the earlier equation

$$-\frac{\partial P}{\partial x} - \rho_E\frac{\partial \phi}{\partial x} + \eta\frac{\partial^2 u}{\partial y^2} = 0 \tag{2.15}$$

with the following

$$-\frac{\partial P}{\partial x} - \rho_E\frac{\partial \phi}{\partial x} + \frac{\partial}{\partial y}\left(\eta\frac{\partial u}{\partial y}\right) = 0, \tag{4.28}$$

which more accurately accounts for the varying viscosity. As a result, we instead obtain the following relation for the slip velocity of the particle

$$u_s = \frac{\epsilon_m kT}{\eta_b ze}\kappa_1(\zeta)\frac{\partial \psi_b}{\partial x} - \frac{k^2T^2\epsilon_m}{\eta_b z^2e^2}\kappa_2(\zeta)\frac{1}{c_b}\frac{\partial c_b}{\partial x}, \tag{4.29}$$

[3]Bazant *et al.* [6] uses the local charge density ρ_E in place of c.

where the coefficient $\kappa_1(\zeta)$ is given by

$$\kappa_1(\zeta) = \int_0^{\zeta} \frac{1}{\eta_s(\phi'')} \left[\int_0^1 \frac{\phi''}{2} \frac{\nu \sinh(\phi'\phi'')}{1 + 2\nu \sinh^2\left(\frac{\phi'\phi''}{2}\right)} \frac{1}{\sqrt{\ln\left(1 + 2\nu \sinh^2 \frac{\phi'\phi''}{2}\right)}} \right.$$
$$\left. \frac{1}{\sqrt{\ln\left(1 + 2\nu \sinh^2 \frac{\phi''}{2}\right)}} d\phi' \right] d\phi''$$
$$= \int_0^{\zeta} \frac{1}{\eta_s(\phi'')} d\phi'', \quad (4.30)$$

recovering the expression obtained by Bazant *et al.* [6], and the coefficient $\kappa_2(\zeta)$ is given by

$$\kappa_2(\zeta) = \int_0^{\zeta} \frac{1}{\eta_s(\phi'')} \left[\int_0^1 \frac{\phi''}{2} \sqrt{\frac{\ln\left(1 + 2\nu \sinh^2 \frac{\phi'\phi''}{2}\right)}{\ln\left(1 + 2\nu \sinh^2 \frac{\phi''}{2}\right)}} d\phi' \right] d\phi'', \quad (4.31)$$

and the viscosity of the fluid is given by $\eta = \eta_s \eta_b$ where η_b is the bulk viscosity assuming a volume fraction of zero for solutes and η_s is the scale factor to include the effects of charge thickening.

Based on our earlier analysis, we can use the following expressions for η_s

$$\eta_s = \left(1 - \frac{0.132 c\nu\pi}{c_{b,\infty}}\right)^{-1.575} \quad (4.32)$$

and

$$\eta_s = \frac{1 + 1.5F(c)}{1 - F(c)} \quad (4.33)$$

where

$$F(c) = \left(\frac{c\nu\pi}{12c_{b,\infty}}\right) + \left(\frac{c\nu\pi}{12c_{b,\infty}}\right)^2 + \left(\frac{c\nu\pi}{12c_{b,\infty}}\right)^3 - 2.3\left(\frac{c\nu\pi}{12c_{b,\infty}}\right)^4. \quad (4.34)$$

These expressions correspond to Equations (4.2), (4.23) and (4.25) respectively, and use 0.63 as the value for the maximum volume fraction.

Again, we can consider a dimensionless version of Equation (4.29) by introducing the characteristic velocity

$$U^* = \frac{\epsilon_m \varphi_T^2}{\eta_b R}. \tag{4.35}$$

With this scaling, we find that the dimensionless slip velocity u_s at the surface of the particle is

$$u_s = \kappa_1(\zeta)\frac{\partial \psi_b}{\partial x} - \frac{\kappa_2(\zeta)}{c_b}\frac{\partial c_b}{\partial x}. \tag{4.36}$$

In the bulk solution, we replace η by $\eta_s \eta_b$ in the momentum conservation equation (2.66).

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

Asymmetric diffusivities and ionic sizes

5.1 Asymmetric diffusivities

In the case where the diffusivities of the cations and anions are different, the Nernst-Planck relation in Equation (2.22) should be written as

$$\begin{cases} \mathbf{j}^+ = -D_+ \nabla c_b - \dfrac{zeD_+}{kT} c_b \nabla \Psi_b + c_b \mathbf{u}, \\ \\ \mathbf{j}^- = -D_- \nabla c_b + \dfrac{zeD_-}{kT} c_b \nabla \Psi_b + c_b \mathbf{u}. \end{cases} \tag{5.1}$$

Scaling the flux by the quantity

$$J_+ = \frac{D_+ c_{b,\infty}}{R}, \tag{5.2}$$

we obtain the equivalent dimensionless relation

$$\begin{cases} \mathbf{j}^+ = -\nabla c_b - c_b \nabla \Psi_b + \mathrm{Pe}_+ c_b \mathbf{u}, \\ \\ \mathbf{j}^- = -\gamma \nabla c_b + \gamma c_b \nabla \Psi_b + \mathrm{Pe}_+ c_b \mathbf{u}, \end{cases} \tag{5.3}$$

where Pe_+ is the Péclet number defined for the cation

$$\mathrm{Pe}_+ = \frac{U^* R}{D_+} = \frac{\epsilon_m \varphi_T^2}{\eta D_+} \tag{5.4}$$

and γ is the ratio of the two diffusivities

$$\gamma = \frac{D_-}{D_+}. \tag{5.5}$$

The conservation of counterions and coions are expressed by the relations

$$\begin{cases} \mathrm{div}(\mathbf{j}^+) = -\nabla^2 c_b - \mathrm{div}(c_b \nabla \Psi_b) + \mathrm{Pe}_+ \mathrm{div}(c_b \mathbf{u}) = 0, \\ \\ \mathrm{div}(\mathbf{j}^-) = -\gamma \nabla^2 c_b + \gamma \mathrm{div}(c_b \nabla \Psi_b) + \mathrm{Pe}_+ \mathrm{div}(c_b \mathbf{u}) = 0, \end{cases} \tag{5.6}$$

which can be combined to yield, in dimensionless form,

$$\begin{cases} (1+\gamma)\nabla^2 c_b + (1-\gamma)\mathrm{div}(c_b \nabla \Psi_b) - 2\mathrm{Pe}_+ \mathrm{div}(c_b \mathbf{u}) = 0, \\ \\ (1-\gamma)\nabla^2 c_b + (1+\gamma)\mathrm{div}(c_b \nabla \Psi_b) = 0. \end{cases} \tag{5.7}$$

When $\gamma = 1$, the equation reduces to the more familiar form (2.27).

The boundary conditions (2.35) will also need to be modified to the following expressions

$$\begin{cases} \dfrac{\partial c_b}{\partial n} = -\dfrac{1}{2}\mathrm{div}_S\left((w + p_1 \mathrm{Pe}_+)\nabla_S \ln c_b + (q + p_2 \mathrm{Pe}_+)\dfrac{ze}{kT}\nabla_S \Psi_b \right), \\ \\ \dfrac{zec_b}{kT}\dfrac{\partial \Psi_b}{\partial n} = -\dfrac{1}{2}\mathrm{div}_S\left((q + p_3 \mathrm{Pe}_+)\nabla_S \ln c_b + (w + p_4 \mathrm{Pe}_+)\dfrac{ze}{kT}\nabla_S \Psi_b \right), \end{cases} \tag{5.8}$$

where the new expressions for p_i for the steric model are given by

$$p_1 = \int_0^{+\infty} \left[\left\{ c_+(y) + \frac{1}{\gamma} c_-(y) \right\} \kappa(\zeta, \phi(y)) - (1 + \frac{1}{\gamma}) c_b \kappa(\zeta, 0) \right] dy, \tag{5.9}$$

$$p_2 = \int_0^{+\infty} -\left[\left\{ c_+(y) + \frac{1}{\gamma} c_-(y) \right\} (\zeta - \phi(y)) - (1 + \frac{1}{\gamma}) c_b \zeta \right] dy, \tag{5.10}$$

$$p_3 = \int_0^{+\infty} \left[\left\{ c_+(y) - \frac{1}{\gamma} c_-(y) \right\} \kappa(\zeta, \phi(y)) \right] dy, \tag{5.11}$$

$$p_4 = \int_0^{+\infty} -\left[\left\{ c_+(y) - \frac{1}{\gamma} c_-(y) \right\} (\zeta - \phi(y)) \right] dy, \tag{5.12}$$

and the new expressions for p_i for the dielectric decrement model are given by

$$p_1 = \int_0^{+\infty} \left[\left\{ c_+(y) + \frac{1}{\gamma} c_-(y) \right\} \kappa_{D2}(\zeta, \phi(y)) - (1 + \frac{1}{\gamma}) c_b \kappa_{D2}(\zeta, 0) \right] dy, \tag{5.13}$$

$$p_2 = \int_0^{+\infty} -\left[\left\{ c_+(y) + \frac{1}{\gamma} c_-(y) \right\} \kappa_{D1}(\zeta, \phi(y)) - (1 + \frac{1}{\gamma}) c_b \kappa_{D1}(\zeta, 0) \right] dy, \tag{5.14}$$

$$p_3 = \int_0^{+\infty} \left[\left\{ c_+(y) - \frac{1}{\gamma} c_-(y) \right\} \kappa_{D2}(\zeta, \phi(y)) \right] dy, \tag{5.15}$$

$$p_4 = \int_0^{+\infty} -\left[\left\{ c_+(y) - \frac{1}{\gamma} c_-(y) \right\} \kappa_{D1}(\zeta, \phi(y)) \right] dy. \tag{5.16}$$

When $\gamma = 1$, we recover the original boundary conditions with no modifications.

Based on empirical data [123, 124, 125], the diffusivities of most aqueous ions are on the order of 1×10^{-9} $\mathrm{m^2 s^{-1}}$, with the exception of hydrogen (H^+) and hydroxide (OH^-) ions, which have about 10 times and 5 times the diffusivity respectively. As such, as long as one is not dealing with strong aqueous acids or bases, the value of γ can be assumed to be in the neighborhood of 0.3 to 3.

5.2 Asymmetric ionic sizes

Suppose the cation has a diameter of a_+ and the anion has a diameter of a_-. The chemical potentials of each ion are then written as

$$\begin{cases} \mu_+ = ze\phi + kT \ln\left(\dfrac{c_+}{1 - a_+^3 c_+ - a_-^3 c_-}\right), \\ \\ \mu_- = -ze\phi + kT \ln\left(\dfrac{c_-}{1 - a_+^3 c_+ - a_-^3 c_-}\right). \end{cases} \tag{5.17}$$

Equating each chemical potential to the chemical potential in the bulk

$$\mu_b = kT \ln\left(\frac{c_{b,\infty}}{1 - a_+^3 c_{b,\infty} - a_-^3 c_{b,\infty}}\right), \tag{5.18}$$

we obtain

$$\begin{cases} \dfrac{c_+}{1 - a_+^3 c_+ - a_-^3 c_-} = \dfrac{c_{b,\infty} \exp\left(-\dfrac{ze\phi}{kT}\right)}{1 - a_+^3 c_{b,\infty} - a_-^3 c_{b,\infty}}, \\ \\ \dfrac{c_-}{1 - a_+^3 c_+ - a_-^3 c_-} = \dfrac{c_{b,\infty} \exp\left(\dfrac{ze\phi}{kT}\right)}{1 - a_+^3 c_{b,\infty} - a_-^3 c_{b,\infty}}, \end{cases} \tag{5.19}$$

and thus

$$\begin{cases} \dfrac{-a_+^3 c_+}{1 - a_+^3 c_+ - a_-^3 c_-} = \dfrac{-a_+^3 c_{b,\infty} \exp\left(-\dfrac{ze\phi}{kT}\right)}{1 - a_+^3 c_{b,\infty} - a_-^3 c_{b,\infty}}, \\ \\ \dfrac{-a_-^3 c_-}{1 - a_+^3 c_+ - a_-^3 c_-} = \dfrac{-a_-^3 c_{b,\infty} \exp\left(\dfrac{ze\phi}{kT}\right)}{1 - a_+^3 c_{b,\infty} - a_-^3 c_{b,\infty}}. \end{cases} \tag{5.20}$$

Hence,

$$\frac{1}{1 - a_+^3 c_+ - a_-^3 c_-} = 1 + \frac{a_+^3 c_{b,\infty} \exp\left(-\dfrac{ze\phi}{kT}\right) + a_-^3 c_{b,\infty} \exp\left(\dfrac{ze\phi}{kT}\right)}{1 - a_+^3 c_{b,\infty} - a_-^3 c_{b,\infty}}, \tag{5.21}$$

and thus

$$c_{\pm} = \frac{c_b \exp\left(\mp \frac{ze\phi}{kT}\right)}{1 - a_+^3 c_{b,\infty} - a_-^3 c_{b,\infty} + a_+^3 c_{b,\infty} \exp\left(-\frac{ze\phi}{kT}\right) + a_-^3 c_{b,\infty} \exp\left(\frac{ze\phi}{kT}\right)}, \tag{5.22}$$

scaling the local concentration by the concentration at the bulk/double layer interface.

In the small ϕ limit (assuming ϕ is positive), steric effects are negligible and we obtain the limit

$$c_{\pm} = c_b \exp\left(\mp \frac{ze\phi}{kT}\right). \tag{5.23}$$

In the large ϕ limit, since $\exp\left(-\frac{ze\phi}{kT}\right)$ is negligible, we can simplify Equation (5.22) to obtain

$$c_{\pm} = \frac{c_b \exp\left(\mp \frac{ze\phi}{kT}\right)}{1 - a_+^3 c_{b,\infty} - a_-^3 c_{b,\infty} + 2a_-^3 c_{b,\infty} \cosh\left(\frac{ze\phi}{kT}\right)}. \tag{5.24}$$

Taking $\nu_- = 2a_-^3 c_{b,\infty}$ and $\nu_+ = 2a_+^3 c_{b,\infty}$, we can rewrite the above relation as

$$c_{\pm} = \frac{c_b \exp\left(\mp \frac{ze\phi}{kT}\right)}{1 - \frac{\nu_+ + \nu_-}{2} + \nu_- \cosh\left(\frac{ze\phi}{kT}\right)}, \tag{5.25}$$

although the ν_+ term does not have a large impact since the large hyperbolic cosine term dominates.

In the limit $\nu_+ \approx \nu_-$, we obtain the exact relation

$$c_{\pm} = \frac{c_b \exp\left(\mp \frac{ze\phi}{kT}\right)}{1 + \nu\left(\cosh\left(\frac{ze\phi}{kT}\right) - 1\right)}. \tag{5.26}$$

Kilic [61] noted that a single $\nu = \nu_-$ (or ν_+ for negatively charged particles) is sufficient to account for steric effects as a first approximation due to the small-potential

and large-potential asymptotic behavior described above. In addition, he demonstrated that the adoption of two different steric parameters had little effect on the final excess charge in the double layer. As such, the single ν model will suffice for our purposes. Also, Kilic noted that a more rigorous analysis would require more complicated expressions for the chemical potential such as the Boublik-Mansouri-Carnahan-Stirling-Leland relation. Doing so would result in more complicated modifications to the Poisson-Boltzmann relation.

Chapter 6

Results and discussion

6.1 Numerical model

In this study, a pseudo-spectral method was used to solve the model equations. Chu and Bazant [3] used the same numerical method to perform the numerical evaluation of the ionic concentration and electric potential fields around a conducting sphere. Following their study, we employed a tensor product of a uniformly spaced grid in the azimuthal direction and a semi-infinite rational Chebyshev grid in the radial direction to define the computational grid. In addition, we used standard Newton iterations to solve the non-linear equations involved in the model. In our numerical simulations, we simultaneously solve for the concentration, electric potential and velocity fields everywhere in the bulk solution to obtain the electrophoretic mobility of the particle. All the simulations were performed using MATLAB.

6.1.1 Physical parameters

The model described in the last few Chapters is governed by six dimensionless physical parameters. The first of them is the dimensionless charge of the particle Q, defined by Equation (3.25). Using the Debye description of the EDL, we can relate the

dimensionless particle charge to the uniform equilibrium zeta potential of the particle

$$Q \sim 2\zeta_0. \tag{6.1}$$

In this relation, ζ_0 is scaled by the thermal voltage, which is 25 mV for a monovalent solution. In other words, a particle with a nondimensional charge of 1 has an equivalent electric potential of 50 mV. The second parameter is the dimensionless applied electric field E. The characteristic electric field is defined here as

$$E^* = \frac{\varphi_T}{R} \tag{6.2}$$

and therefore depends on the size of the particle. For a 1 μm particle, the characteristic electric field is $E^* = 250$ V/cm. The third parameter is the ratio ϵ of the Debye length λ_D to the particle radius R. The Debye length is defined in Equation (2.6), and is essentially governed by the bulk concentration $c_{b,\infty}$. For a concentrated monovalent aqueous electrolyte solution at 25°C where $c_{b,\infty} = 0.18$ M, the Debye length is 0.72 nm. Thus, for a 1 μm particle, $\epsilon \approx 1.0 \times 10^{-3}$. For an aqueous monovalent electrolyte solution at 25°C where $c_{b,\infty} = 1.0 \times 10^{-3}$ M, the Debye length is $\lambda_D = 9.7$ nm, which results in $\epsilon \approx 1.0 \times 10^{-2}$ for a 1 μm particle. The fourth parameter is the Péclet number Pe, which is defined in Equation (2.25). For an ionic diffusivity $D = 10^{-9}$ m^2/s, Pe is approximately 0.45. The fifth parameter is the packing parameter $\nu = 2a^3 c_{b,\infty}$ describing crowding effects, which depends on the ionic concentration of the solution and the effective ionic size. ν varies between 1.0×10^{-5} for dilute solutions with small ions and 0.1 for highly concentrated solutions with large ions. The final parameter is the dielectric decrement parameter α, which depends on the ionic concentration of the solution and the nature of ion-solvent interactions involved, and has a characteristic scale defined in Equation (3.4). α varies between 0.001 for dilute solutions with weaker ion-solvent interactions and 0.3 for concentrated solutions with stronger ion-solvent interactions.

6.2 Weakly non-linear regime

Before we discuss the results of the numerical simulations proper, we briefly focus on the case of weak applied electric fields to compare this study to the analytical expression found by Yariv [39] for the electrophoretic mobility of ideally polarizable particles. In this case, we can adopt the limit of zero surface conduction and assume the absence of steric effects. By neglecting crowding effects, we recover the Gouy-Chapman model

$$Q = \int_0^{\pi} 2 \sinh \frac{\zeta}{2} \sin \theta d\theta. \tag{6.3}$$

By using this relation in conjunction with the earlier expression for the zeta potential (3.24), we can obtain an expression for the particle potential Ψ_P in a similar form to that obtained by Yariv [39]

$$\Psi_P = 2 \sinh^{-1} \left(\frac{3QE}{16 \sinh \frac{3E}{4}} \right). \tag{6.4}$$

The numerical simulations performed with our model demonstrate convergence towards the results of Yariv when $\nu = 0$ for different values of Q as evidenced in Figs. 6-1 and 6-2.

6.3 Steric effects

6.3.1 Basic model

Figs. 6-1 and 6-2 demonstrate that the mobility of a charged polarizable spherical particle is highly dependent on the packing parameter ν of the particle-electrolyte configuration. This trend is markedly significant at large values of E.

At low values of Q and E, steric effects are relatively insignificant, as shown in Fig. 6-1. Hence, the mobility values for all the values of ν considered are similar, and are well predicted by the model developed by Yariv [39] describing the weakly non-linear regime. This similarity breaks down for increasing Q, as shown in Fig.

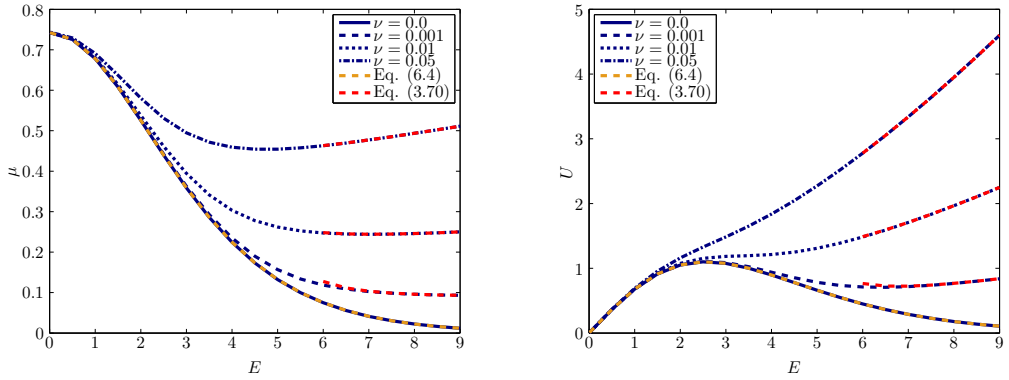

Figure 6-1: Dimensionless electrophoretic mobility (left) and velocity (right) of a positively charged particle as a function of the dimensionless applied electric field for different values of the packing parameter ν. The dimensionless charge of the particle is $Q = 1$. Surface conduction is neglected in these simulations ($\epsilon = 0$). The asymptotic approximation (3.70) is plotted in red dashed lines, while the the analytic model of Yariv (6.4) is plotted in orange dashed lines.

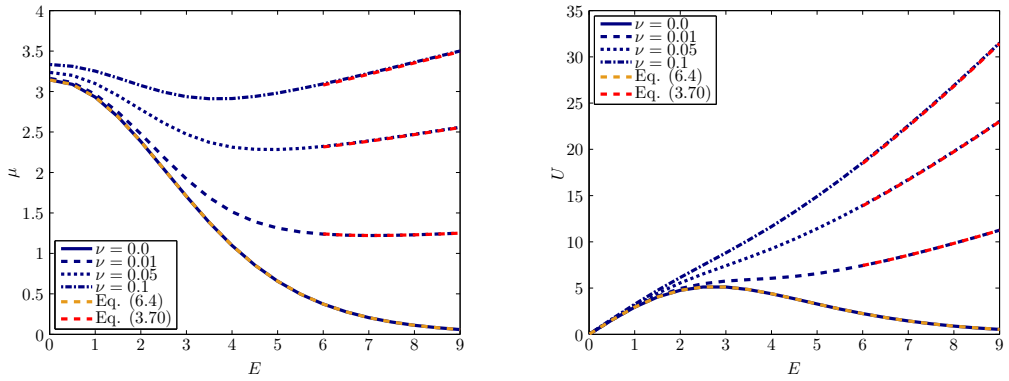

Figure 6-2: Dimensionless electrophoretic mobility (left) and velocity (right) of a positively charged particle as a function of the dimensionless applied electric field for different values of the packing parameter ν. The dimensionless charge of the particle is $Q = 5$. Surface conduction is neglected in these simulations ($\epsilon = 0$). The asymptotic approximation (3.70) is plotted in red dashed lines, while the the analytic model of Yariv (6.4) is plotted in orange dashed lines.

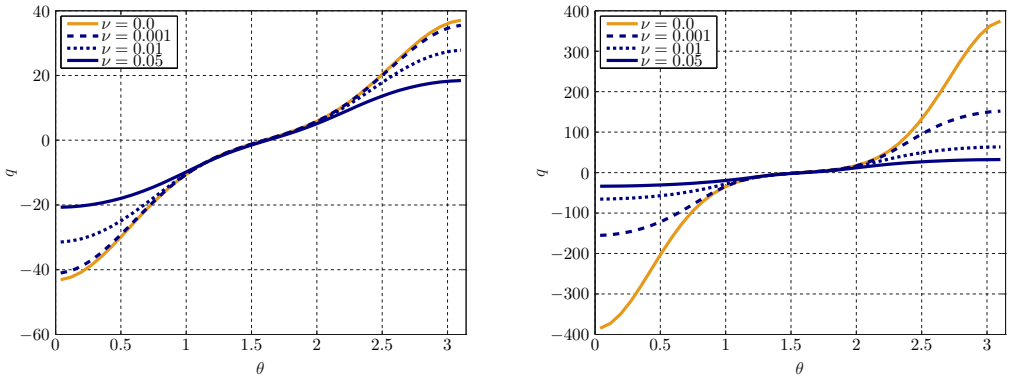

Figure 6-3: Dimensionless surface excess charge for distinct values of the packing parameter ν. The dimensionless applied electric field is $E = 4$ (left) and $E = 7$ (right). In these simulations, $Q = 1$ and $\epsilon = 0$. The polarization of the particle can be observed in both figures. The surface charge diverges with increasing applied electric field when steric effects are neglected ($\nu = 0$).

6-2, even at low applied electric fields. Physically, since the zeta potential increases at higher values of Q, the extent to which the solvated ions are packed around the particle influences their spatial distribution around the particle more strongly.

At high values of E, ζ increases significantly near the poles of the particle, and steric effects heavily influence the electrophoretic mobility. In the absence of surface conduction ($\epsilon \to 0$), an asymptotic expression (3.70) can be derived for the mobility, and the numerical simulations show good agreement with this expression, as evidenced in Figs. 6-1 and 6-2. A physical account of this expression can be offered as follows: when steric effects are neglected, the surface charge of the particle diverges with increasing electric fields, and as shown in Fig. 6-3, the zeta potential distribution around the particle is relatively symmetric, resulting in a low electrophoretic mobility. On the other hand, for large values of ν, the distribution of solvated ions around the particle is affected by steric effects due to the finite size of the ions in the EDL, causing an asymmetry in the zeta potential distribution and thus a finite electrophoretic mobility.

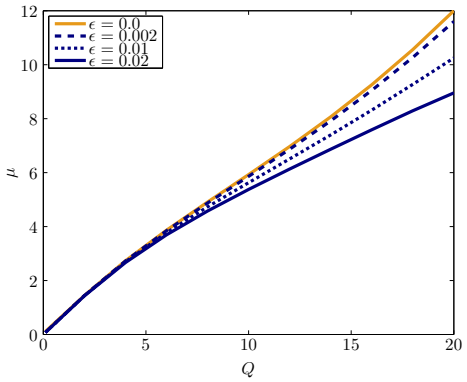

Figure 6-4: Dimensionless electrophoretic mobility of a positively charged particle as a function of the particle charge for distinct values of the ratio ϵ of the Debye length to the particle radius. The dimensionless electric field was held fixed at a value $E = 0.5$, and the packing parameter was held fixed at a value $\nu = 0.1$. We observe that electrophoretic mobility decreases with increased surface conduction, as shown by Khair *et al.* for the case of fixed surface charge particles [45].

6.3.2 Surface conduction

Our numerical simulations also suggest that electrophoretic mobility decreases with increased surface conduction, as shown in Fig. 6-4 for the case of particles with increasing global charge. At low values of Q and hence ζ, the excess charge q and excess ion concentration w in the EDL remain small. Hence, according to Equation (2.57) describing ionic species conservation across the bulk/EDL interface, surface conduction is relatively insignificant. Since the particle does not significantly distort the concentration field in its vicinity, concentration polarization remains negligible, as shown in Fig. 6-5. In addition, Fig. 6-6 shows that the contribution of the diffusio-osmotic flow to the slip velocity is practically zero when the charge of the studied particle remains small.

At higher values of Q, the excess charge q and excess ionic concentration w increase. Hence, surface conduction becomes significant in the EDL. As a consequence, the concentration field becomes polarized in the immediate vicinity of the particle. This physical effect is shown in Fig. 6-5, and is enhanced at high values of the parameter ϵ relating the thickness of the EDL to the radius of the particle. We can also see from Fig. 6-6 that concentration polarization results in significant diffusio-

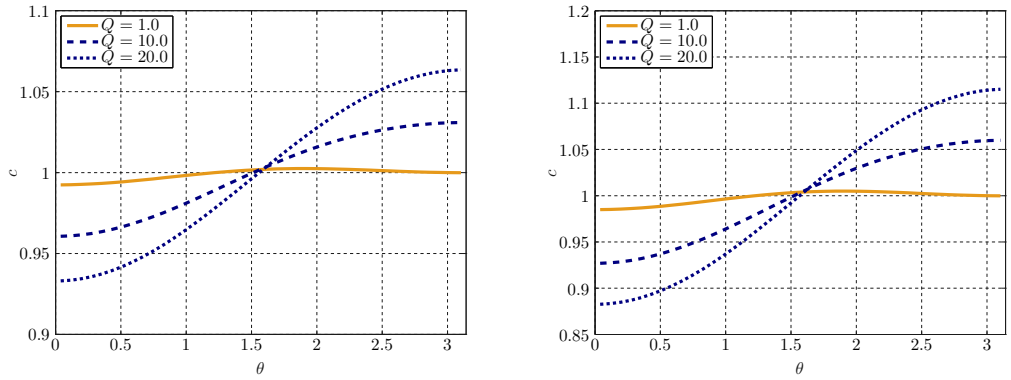

Figure 6-5: Concentration polarization in the immediate vicinity of a charged particle for $\epsilon = 0.01$ (left) and $\epsilon = 0.02$ (right). The simulation parameters here are $E = 0.5$ and $\nu = 0.1$.

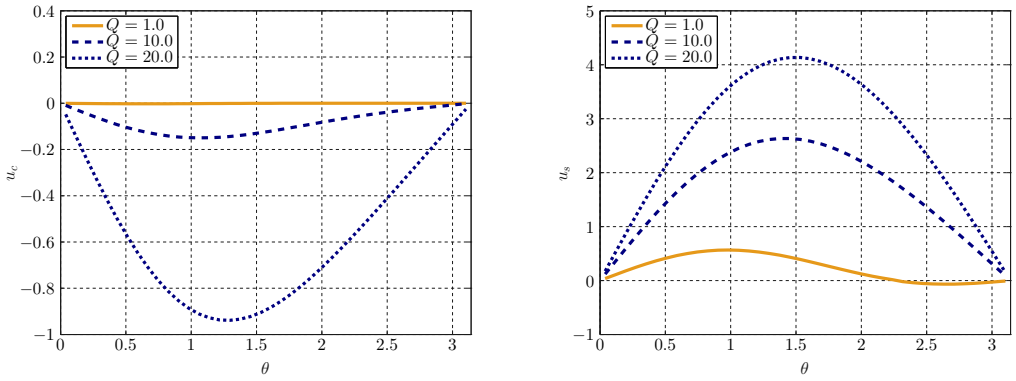

Figure 6-6: Diffusio-osmotic flow (left) and slip velocity (right) at the surface of a charged particle for $\epsilon = 0.02$. The simulation parameters here are $E = 0.5$ and $\nu = 0.1$. For highly charged particles, significant diffusio-osmotic flow can be observed (left), reducing the total slip at the surface of the particle (right).

osmotic flow, which in turn reduces the slip velocity at the surface of the particle. As a consequence, the net electrophoretic motion of the particle is significantly reduced.

6.3.3 Advection and high fields

We extend our analysis to high applied electric fields and observe that surface conduction again results in a decrease in electrophoretic mobility, as shown in Fig. 6-7. In this case, the zeta potential reaches high values on the particle surface. This results in significant surface conduction, as evidenced by the values of the excess salt concentration w plotted in Fig. 6-8. (Recall that w is proportional to the Dukhin number

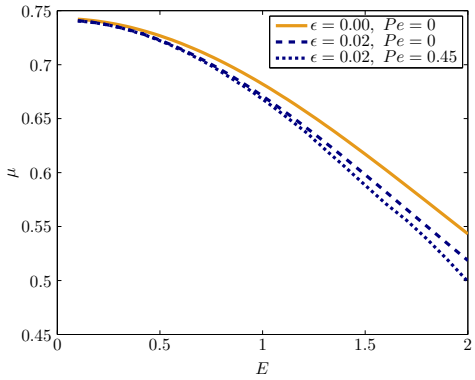

Figure 6-7: Dimensionless electrophoretic mobility of a positively charged particle as a function of the dimensionless applied electric field for distinct values of the ratio ϵ and of the Péclet number. The packing parameter $\nu = 0.015$ corresponds to a bulk concentration $c_{b,\infty} = 0.1M$ for an ionic diameter $a = 0.5$ nm. The Debye length is thus $\lambda_D = 0.97$ nm. For $\epsilon = 0.02$, the particle radius is $R = 49$ nm, giving a characteristic electric field of approximately $E^* = 5 \times 10^3$ V/cm.

Du, which gives a measure of surface conduction.) This in turn enhances concentration polarization and diffusio-osmotic flow. However, in this case, the diffusio-osmotic flow contribution to the slip velocity is much smaller than the contribution of electro-osmotic flow, as shown in Fig. 6-8 as well.

From Equation (2.27), we expect that concentration gradients enhance advective transport in the bulk solution, thereby significantly reducing ionic concentration fluctuations in the electric double layer. This result corroborates with the asymptotic analysis performed by Schnitzer and Yariv [34], who demonstrated for dielectric particles that the concentration remains constant at first order in the bulk solution in the limit Pe $\gg 1$. The ionic concentration field in the vicinity of the particle surface is shown in Fig. 6-9 when advective transport is neglected and when the Péclet number takes the typical value of 0.45. However, this effect is outweighed by the dramatic increase in surface fluxes in the double layer due to advection. The increase in surface flux biases the chemiphoretic flow in one direction while keeping the electroosmotic flow and effective surface conduction constant, causing a further reduction in the electrophoretic mobility.

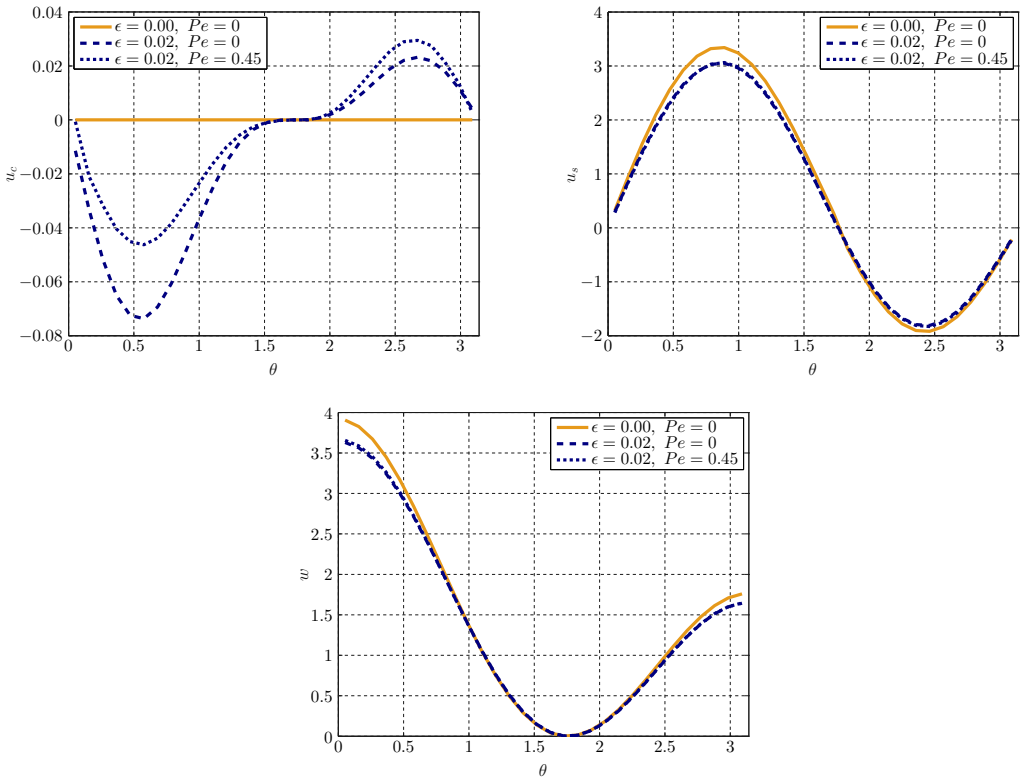

Figure 6-8: Diffusio-osmotic flow (upper left), slip velocity (upper right) and excess salt concentration (bottom) at the surface of a charged particle for $Q = 1$, $E = 1.5$, and distinct values of ϵ and Pe. The packing parameter is $\nu = 0.015$, which corresponds to a bulk concentration $c_{b,\infty} = 0.1M$ for an ionic diameter $a = 0.5$ nm. The Debye length is thus $\lambda_D = 0.97$ nm. For $\epsilon = 0.02$, the particle radius is $R = 49$ nm, giving a characteristic electric field of approximately $E^* = 5 \times 10^3$ V/cm.

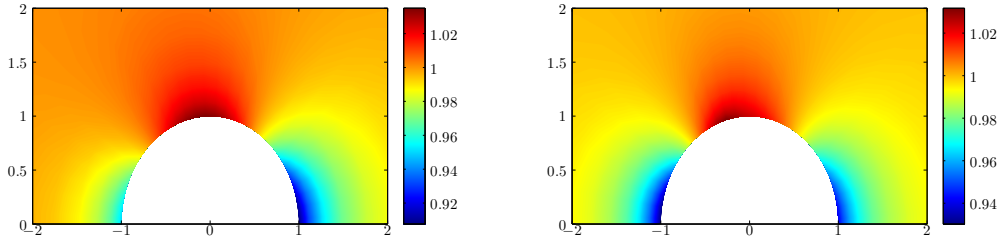

Figure 6-9: Ionic concentration field in the bulk solution for Pe $= 0$ (left) and Pe $= 0.45$ (right). The simulation parameters here are $Q = 1$, $E = 1.5$ and $\epsilon = 0.02$. The packing parameter is $\nu = 0.015$, which corresponds to a bulk concentration $c_{b,\infty} = 0.1M$ for an ionic diameter $a = 0.5$ nm. The Debye length is thus $\lambda_D = 0.97$ nm. For $\epsilon = 0.02$, the particle radius is $R = 49$ nm, giving a characteristic electric field of approximately $E^* = 5 \times 10^3$ V/cm. Concentration gradients enhance advective transport in the bulk solution, thereby significantly reducing ionic concentration fluctuations in the EDL.

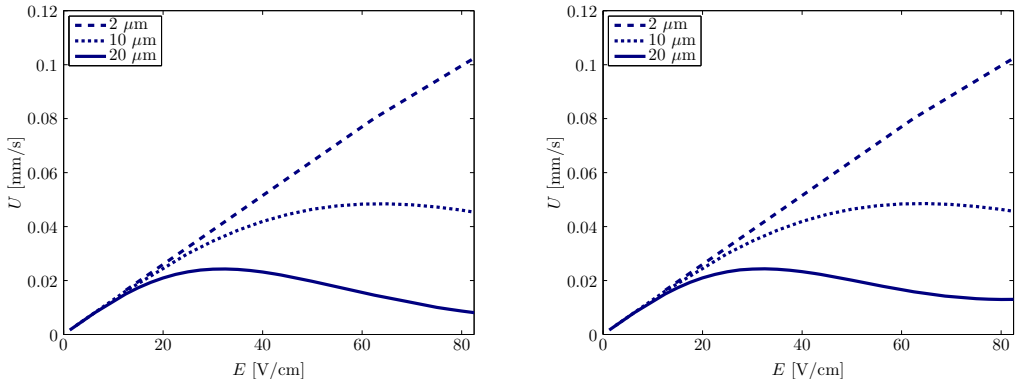

Figure 6-10: Electrophoretic speeds of positively charged ($Q = 1$) ideally polarizable particles of different sizes as a function of the applied electric field for $\nu = 6.02 \times 10^{-5}$ (left) and $\nu = 4.82 \times 10^{-4}$ (right) to simulate the steric effects of sodium (diameter 100 pm) and chloride (200 pm) ions respectively. All the axes reflect dimensional quantities. The dimensional bulk concentration used was 0.05 M (giving a Debye length of 1.37 nm), and the typical Péclet number of 0.45 was adopted.

6.3.4 Dimensional graphs for experimental validation

We intend to validate our numerical simulations in the future through experiments with both conducting particles and dielectric particles. In particular, we hope to be able to perform the separation of particles of different sizes using large electric fields. In this subsection, we plot the variation of the dimensional electrophoretic speed of particles of different sizes with the dimensional applied electric field, along with different physical parameters, to seek the optimal experimental conditions to achieve maximal separation.

Ideally polarizable particles

In Figs. 6-10 and 6-11, we investigate the effects of ionic size on particle separation. As suggested by Fig. 6-10, if typical ionic sizes are used to compute the value of ν, the nature of the electrolyte does not have a significant impact on the electrophoretic speeds of the particles. However, as suggested by Fig. 6-11, if a larger ion correlation length is instead used to compute the value of ν, we see that significantly different trends for the electrophoretic speeds are obtained. The experiments could thus potentially offer us a handle to determine the appropriate ionic correlation length to use

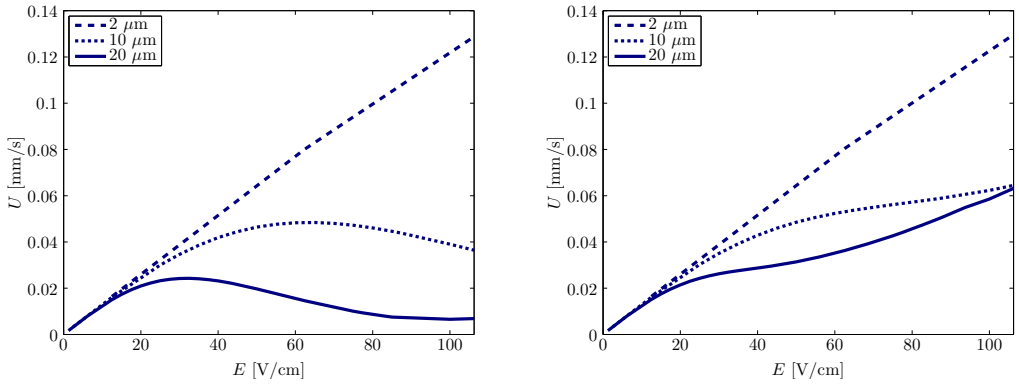

Figure 6-11: Electrophoretic speeds of positively charged ($Q = 1$) ideally polarizable particles of different sizes as a function of the applied electric field for $\nu = 6.02 \times 10^{-5}$ (left) and $\nu = 2.06 \times 10^{-2}$ (right) to simulate the steric effects of sodium ions (diameter 100 pm) and of ions with a correlation length of 700 pm respectively. All the axes reflect dimensional quantities. The dimensional bulk concentration used was 0.05 M (giving a Debye length of 1.37 nm), and the typical Péclet number of 0.45 was adopted.

to compute ν for different ions.

In Fig. 6-12, we investigate the effects of particle charge on particle separation. Since electrophoretic speed scales approximately with the charge of the particle, higher charges give us higher absolute velocity differences and thus better separation. It would thus be ideal to maximize the charge of the particles in the experiments.

In Fig. 6-13, we investigate the effects of the bulk concentration on particle separation. We observe that changing the bulk concentration does not significantly affect the electrophoretic speeds of the particles, suggesting that increasing or decreasing the bulk concentration of the ions in the electrolyte may not be a good way to optimize the particle separation. (Increasing the bulk concentration decreases the Debye length and thus decreases the relative importance of surface conduction, but also increases the effects of steric packing. Decreased surface conduction increases the absolute velocity differences between the differently-sized particles, but increased steric packing decreases these differences. The net result of a change in the bulk concentration is thus indeed expected to be small.)

In Fig. 6-14, we investigate the effects of particle size on particle separation. Smaller particles have larger electrophoretic speeds, and thus using smaller particles

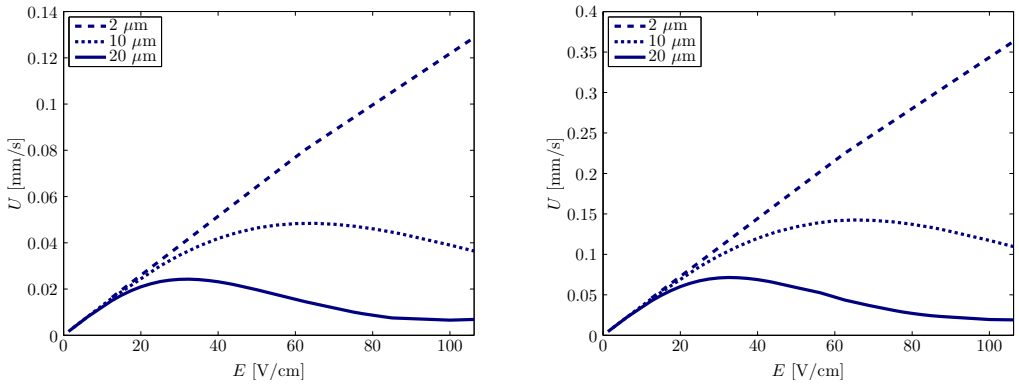

Figure 6-12: Electrophoretic speeds of positively charged ideally polarizable particles of different sizes as a function of the applied electric field for $Q = 1$ (left) and $Q = 3$ (right). All the axes reflect dimensional quantities. The dimensional bulk concentration used was 0.05 M (giving a Debye length of 1.37 nm), the dimensional ionic diameter used was 100 pm (giving $\nu = 6.02 \times 10^{-5}$), and the typical Péclet number of 0.45 was adopted.

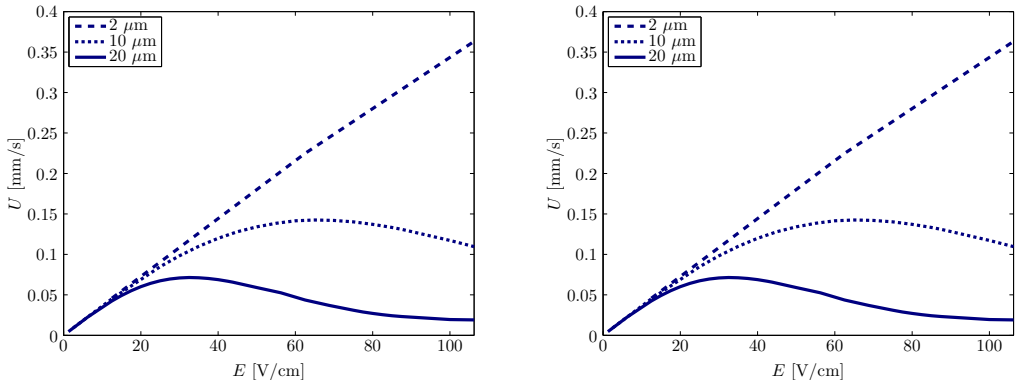

Figure 6-13: Electrophoretic speeds of positively charged ($Q = 3$) ideally polarizable particles of different sizes as a function of the applied electric field for bulk concentrations of 0.05 M (left) and 0.3 M (right). These correspond to Debye lengths of 1.37 nm and 0.56 nm respectively. All the axes reflect dimensional quantities. The dimensional ionic diameter used was 100 pm (giving $\nu = 6.02 \times 10^{-5}$ for the left figure and $\nu = 3.61 \times 10^{-4}$ for the right figure), and the typical Péclet number of 0.45 was adopted.

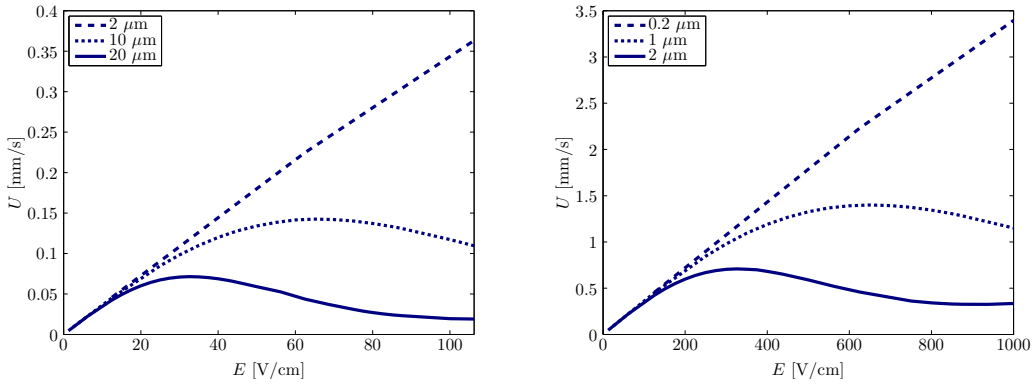

Figure 6-14: Electrophoretic speeds of positively charged ($Q = 3$) ideally polarizable particles of different sizes as a function of the applied electric field for larger (left) and smaller (right) particles. The dimensional bulk concentration used was 0.3 M (giving a Debye length of 0.56 nm), the dimensional ionic diameter used was 100 pm (giving $\nu = 3.61 \times 10^{-4}$), and the typical Péclet number of 0.45 was adopted.

allows us to increase the absolute velocity differences between the particles. However, the non-linear regime for smaller particles begins only at larger electric fields, meaning that we would require larger electric fields in order to obtain observable particle separation. The electric field required for separation scales approximately with the inverse of the particle size.

Based on the above analysis, it appears that the optimum experimental range for ideally polarizable particles would be an electric field strength in the neighborhood of 80 V/cm for particles with diameters between 2 and 20 μm and as highly charged as possible. Smaller particles would require larger field strengths, and conversely, larger particles would require smaller field strengths.

Dielectric particles

Since dielectric particles are experimentally easier to deal with, we decided to run numerical simulations for dielectric particles as well to determine the feasibility of using them to achieve particle separation based on size. To obtain these results, we modified the code such that instead of having a constant electric potential, the particle now has a constant zeta potential everywhere around it instead. This causes the non-linear effects to only be observable at higher particle charges and electric

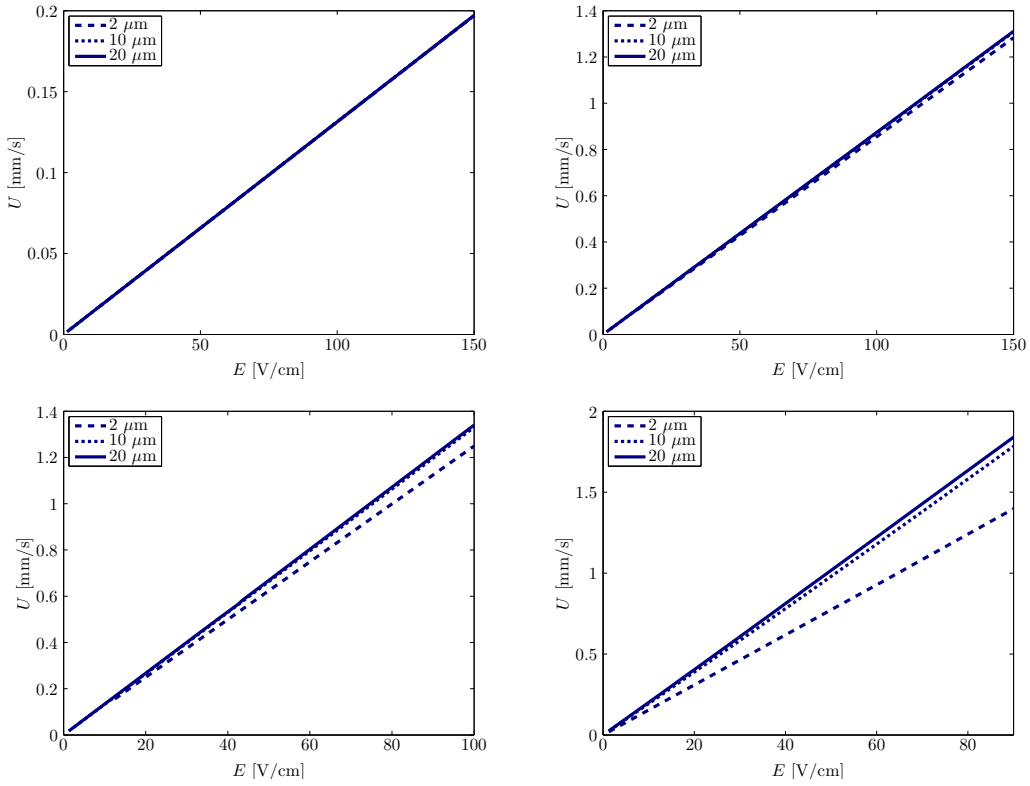

Figure 6-15: Electrophoretic speeds of positively charged dielectric particles of different sizes as a function of the applied electric field for $Q = 1$ (upper left), $Q = 10$ (upper right), $Q = 25$ (lower left) and $Q = 100$ (lower right). (Note the different scales for the horizontal axes in the top and the bottom figures.) The dimensional bulk concentration used was 0.05 M (giving a Debye length of 1.37 nm), the dimensional ionic diameter used was 100 pm (giving $\nu = 6.02 \times 10^{-5}$), and the typical Péclet number of 0.45 was adopted.

fields since the non-linearity in ideally polarizable particles arises from the large zeta potentials near the poles of the particle.

In Fig. 6-15, we investigate the effects of particle charge on particle separation. The higher the particle charge, the higher the zeta potential, and thus the greater the non-linear effects. In the case of particles with sizes between 2 and 20 μm, we observe that particle separation becomes significant above $Q = 25$.

In Fig. 6-16, we investigate the effects of ionic size on particle separation. The difference in speeds due to ionic size differences is not as large as that predicted by the model for ideally polarizable particles, suggesting that experiments using dielectric particles may not be able to give us good predictions for ion correlation lengths.

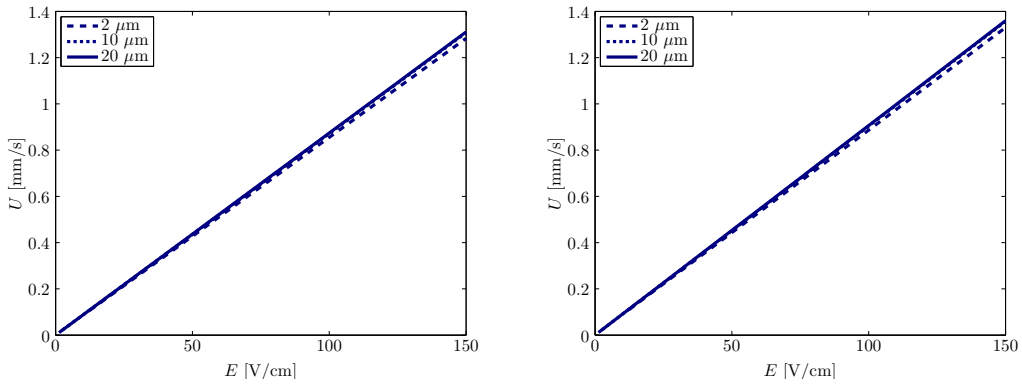

Figure 6-16: Electrophoretic speeds of positively charged ($Q = 10$) dielectric particles of different sizes as a function of the applied electric field for $\nu = 6.02 \times 10^{-5}$ (left) and $\nu = 2.06 \times 10^{-2}$ (right) to simulate the steric effects of sodium ions (diameter 100 pm) and of ions with a correlation length of 700 pm respectively. The dimensional bulk concentration used was 0.05 M (giving a Debye length of 1.37 nm) and the typical Péclet number of 0.45 was adopted.

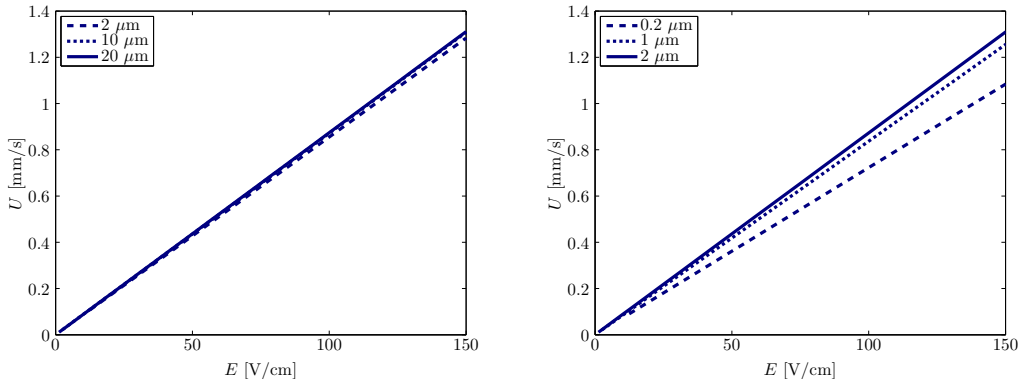

Figure 6-17: Electrophoretic speeds of positively charged ($Q = 10$) dielectric particles of different sizes as a function of the applied electric field for larger (left) and smaller (right) particles. The dimensional bulk concentration used was 0.05 M (giving a Debye length of 1.37 nm), the dimensional ionic diameter used was 100 pm (giving $\nu = 6.02 \times 10^{-5}$), and the typical Péclet number of 0.45 was adopted.

In Fig. 6-17, we investigate the effects of particle size on particle separation. Smaller particles give us larger absolute velocity differences and thus better separation. Unlike in the case of ideally polarizable particles, the use of smaller dielectric particles does not come with the trade-off of higher electric fields; instead, the trade-off comes in the form of lower visibility in imaging equipment.

Based on the above analysis, it appears that the optimum experimental range for dielectric particles would be an electric field strength of above 100 V/cm for particles with diameters smaller than 2 μm and as highly charged as possible. The electric field strength and particle charge can be reduced by using smaller particles, but these particles would be more difficult to observe in a microscope or other imaging equipment. The largest and smallest particles should ideally differ in size by at least two orders of magnitude to obtain more easily observable separation.

6.4 Dielectric decrement

6.4.1 Similarities with steric effects

We observe that the dielectric decrement model results in similar qualitative effects on electrophoretic mobility as the steric-only model. These similarities are present both with and without surface conduction and advection. We present here a set of simulation results that have identical simulation parameters to those of Figs. 6-1 to 6-4 and Fig. 6-8, with the exception that α is varied instead of ν.

In Fig. 6-18, the mobility decays to zero at high E when both ν and α are set to zero, and increases with increasing α especially at high E. These are the same trends we observed earlier in the steric-only model in Figs. 6-1 and 6-2. There is, however, one important difference between the steric-only model and the dielectric decrement model: at low Q and low E, the mobility was observed to be invariant with ν, but we see here a slight dependence with α. This can be explained by the difference in form of the modified Poisson-Boltzmann relation in both models: in the steric-only model, the relation (2.10) predicts negligible steric effects at low potentials, but in the

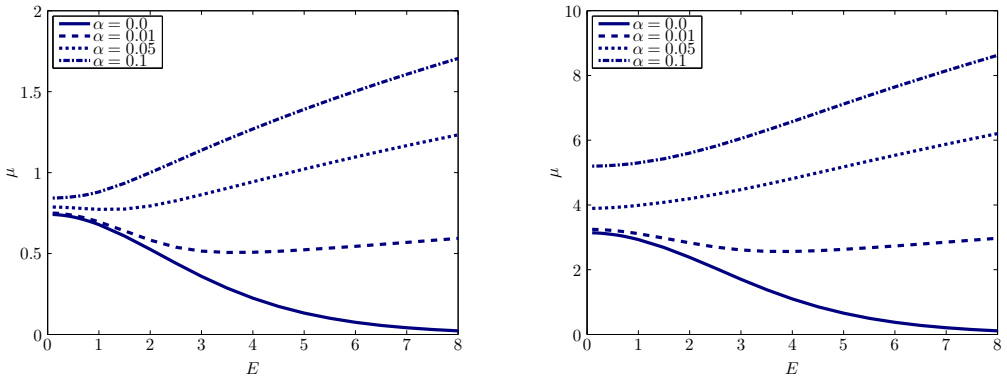

Figure 6-18: Dimensionless electrophoretic mobility of a positively charged particle as a function of the dimensionless applied electric field for different values of the dielectric decrement parameter α. The dimensionless charge of the particle is $Q = 1$ (left) and $Q = 5$ (right). Surface conduction is neglected in these simulations ($\epsilon = 0$) and the packing parameter ν is set to 0. The results in this Figure should be compared with Figs. 6-1 and 6-2.

dielectric decrement model, the relation (3.5) suggests that α continues to affect the charge distribution at low potentials since it affects both the first and second spatial derivatives of the electric potential.

In Fig. 6-19, the charge polarization decreases with increasing α but increases with increasing E. In addition, the point on the sphere with zero excess surface charge moves further away from the centerline ($\theta = \frac{\pi}{2}$) with increasing α, suggesting a higher asymmetry in the charge distribution with increasing α. Again, these are similar trends to those observed in the steric-only model as shown in Fig. 6-3. Observe, however, that for the same numerical values of α and ν, the dielectric decrement effects appear to be stronger than the steric effects. It can be seen from Equations (3.59) and (3.60) that the two models can be compared more fairly if ν scales as 4α.

In Fig. 6-20, the mobility decreases with increasing ϵ in a similar fashion to the mobility in the steric-only model shown in Fig. 6-4. However, the mobilities in the dielectric decrement model appear to increase more strongly with Q than in the steric-only model. An understanding of this trend can be gleaned, at least approximately, from the asymptotic mobility given in Equation (3.58) in the limit of zero surface conduction. Due to the change in dielectric permittivity, the asymptotic mobility is

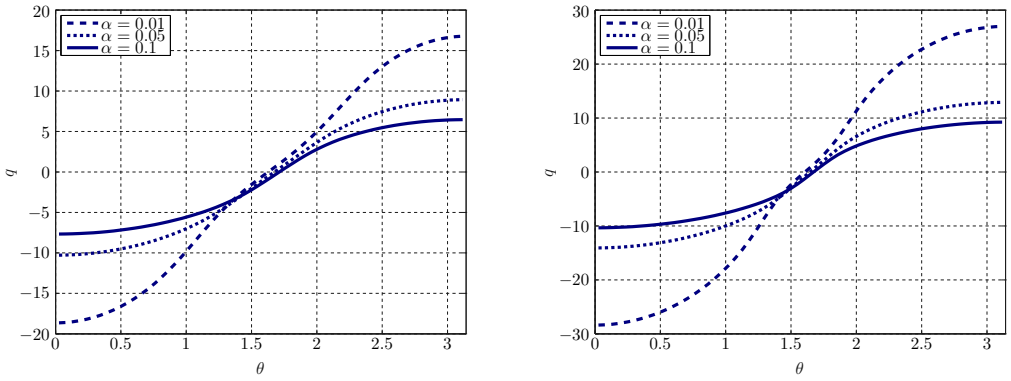

Figure 6-19: Dimensionless surface excess charge for distinct values of the dielectric decrement parameter α. The dimensionless applied electric field is $E = 4$ (left) and $E = 7$ (right). In these simulations, $Q = 1$, $\epsilon = 0$ and $\nu = 0$. The results in this Figure should be compared with Fig. 6-3.

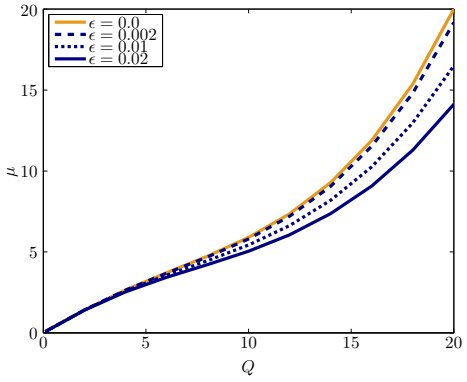

Figure 6-20: Dimensionless electrophoretic mobility of a positively charged particle as a function of the particle charge for distinct values of the ratio ϵ of the Debye length to the particle radius. The dimensionless electric field was held fixed at a value $E = 0.5$, and the dielectric decrement parameter was held fixed at a value $\alpha = 0.025$. The results in this Figure should be compared with Fig. 6-4.

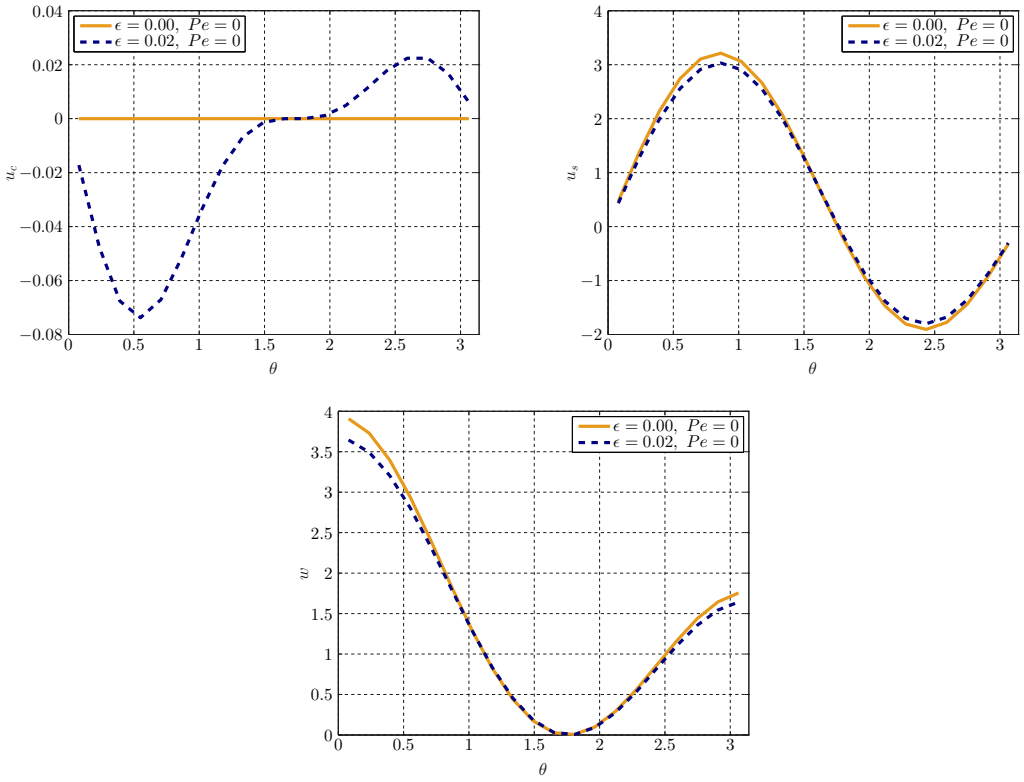

Figure 6-21: Diffusio-osmotic flow (upper left), slip velocity (upper right) and excess salt concentration (bottom) at the surface of a charged particle for $Q = 1$, $E = 1.5$, and distinct values of ϵ. The dielectric decrement parameter is $\alpha = 0.00375$. The results in this Figure should be compared with Fig. 6-8.

larger in the dielectric decrement case, and this increase is modeled by the extra term in the denominator of relation (3.58), which is negative due to the sign of the term c_1 and is indirectly dependent on the zeta potential. An increase in Q increases the average zeta potential around the sphere and amplifies this effect, resulting in larger mobilities.

In Fig. 6-21, we essentially see the same qualitative effects of surface conduction as in the steric-only model (Fig. 6-8). Taken together, Figs. 6-18 to 6-21 suggest that the dielectric decrement model is really an extended steric model that accounts for both ion-solvent interactions as well as ionic concentration saturation near the particle surface. In the limit of extremely thin double layers and zero surface conduction, as well as small particle charges, we can neglect ion-solvent interactions, and it is in this limit that we perform a direct comparison between the dielectric decrement model

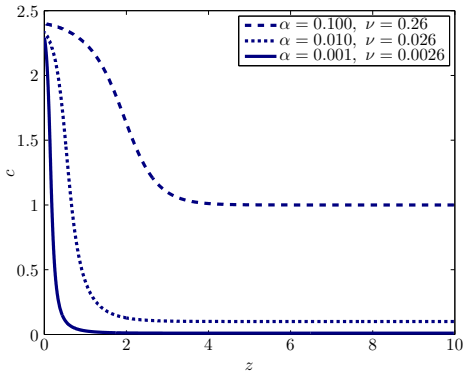

Figure 6-22: Ionic concentration in the EDL for distinct values of ν and α. The parameters used for this simulation are identical to those chosen by Hatlo *et al.* [1] in their depiction of the ionic concentration distribution in the EDL. In particular, the value $\zeta = 15$ was chosen for the zeta potential. The vertical axis is scaled by the bulk concentration at infinity, and the horizontal axis is in terms of dimensionless distance from the particle surface. Observe the saturation of the ionic concentration near the particle surface, especially at high α.

and the steric-only model.

6.4.2 Dielectric decrement as an alternative steric theory at low particle charges

To demonstrate the ability of the dielectric decrement model to simulate steric packing, we plot the variation of ionic concentration with distance from the particle surface in Fig. 6-22. While the nonlinearity of the modified Poisson-Boltzmann relation in the dielectric decrement model causes it to be less well-behaved at higher zeta potentials, we were able to reasonably replicate the results of Hatlo *et al.* [1] in Fig. 6-22 with our numerical simulations at a reasonably high zeta potential value of $\zeta = 15$. Observe that the ionic concentration saturates near the particle surface at a value of approximately $\frac{1}{4\alpha}$, even though this approximation becomes less accurate at smaller α. This lends credence to our suggestion that the dielectric decrement model could possibly account for steric packing on its own. Note, also, that $4\alpha > \nu$ in all the physical configurations in the Figure. As we will demonstrate later, this suggests that dielectric decrement effects rather than steric effects limit the ionic packing near

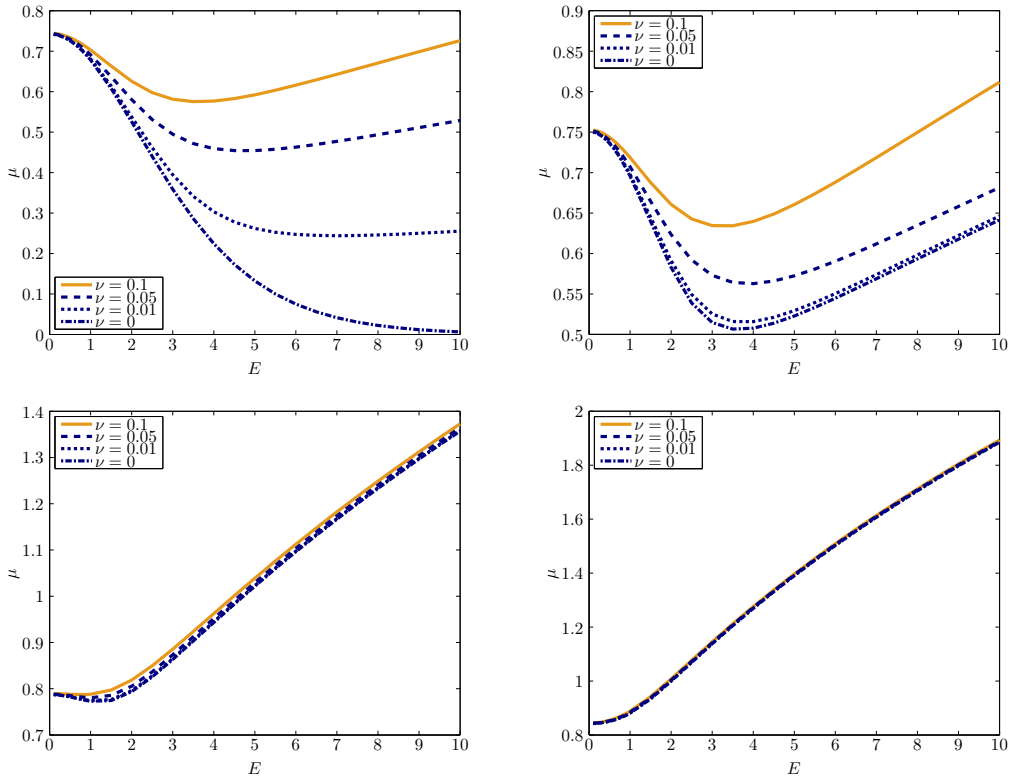

Figure 6-23: Dimensionless electrophoretic mobility of a positively charged particle as a function of the dimensionless applied electric field for different values of the packing parameter ν and of the dielectric decrement parameter α. The dimensionless charge of the particle is $Q = 1$. Surface conduction is neglected in these simulations ($\epsilon = 0$). The parameters used for this simulation are $\alpha = 0$ (upper left), $\alpha = 0.01$ (upper right), $\alpha = 0.05$ (bottom left) and $\alpha = 0.1$ (bottom right).

the particle surface in these physical configurations.

In Fig. 6-23, we plot the variation of electrophoretic mobility over increasing applied electric fields with different ν and α. Observe that some of the lines bunch together, especially at higher α. This suggests that for a fixed value of α, the mobility becomes independent of ν below a critical value of ν. We propose that when this occurs, dielectric decrement effects rather than steric effects limit the ionic packing near the particle surface. At this point, we bring once again to the reader's attention the saturation concentrations predicted by the two models in the absence of surface conduction and at high applied fields, $\rho_{max} \propto \frac{1}{\nu}$ for the steric model as suggested by Equation (3.59), and $\rho_{max} \propto \frac{1}{4\alpha}$ for the dielectric decrement model as suggested by Equation (3.60). In Table (6.1), we demonstrate using these scaling factors for ρ_{max}

	ν	0	0.01	0.05	0.1
α	ρ_{max}	∞	100	20	10
0	∞	N/A	ν	ν	ν
0.01	25	$\boldsymbol{\alpha}$	$\boldsymbol{\alpha}$	ν	ν
0.05	5	$\boldsymbol{\alpha}$	$\boldsymbol{\alpha}$	$\boldsymbol{\alpha}$	$\boldsymbol{\alpha}$
0.1	2.5	$\boldsymbol{\alpha}$	$\boldsymbol{\alpha}$	$\boldsymbol{\alpha}$	$\boldsymbol{\alpha}$

Table 6.1: Table describing the predicted saturation concentration ρ_{max} corresponding to each value of ν and α and the parameter giving the lower concentration for each (ν, α) pair. The entries where the mobility appears to be independent of ν in Fig. 6-23 are bolded.

that the parameter that gives the lower saturation concentration is the parameter that governs steric packing in the electric double layer and thus the parameter that limits the electrophoretic mobility.

There are two ways to account for this: firstly, that either the steric-only model or the dielectric decrement-only model is the more accurate physical model for describing steric packing; or secondly, that either model could be accurate depending on the size and polarizability of the ions involved. Since there is no physical reason for us to believe the first at the moment, we adopt a view that to determine the mobility of the particles in a fluid, the size and polarizability of the ions in the electrolyte should be used to determine the ratio $\sigma = \frac{\nu}{4\alpha}$ and thereby determine if the steric-only model or the dielectric decrement-only model is more suitable to characterize the electrolyte. If $\sigma < 1$, the dielectric decrement-only model is more suitable; if $\sigma > 1$, the steric-only model is more suitable, and if $\sigma \approx 1$, a combination of the two models may be required to adequately account for the physical situation.

6.4.3 Condensed layer approximation

We end this section by evaluating the condensed layer model derived at the end of Chapter 3. Firstly, we perform a direct comparison between our numerical model and our analytic condensed layer model. Figs. 6-24 and 6-25 compare the electric potential and charge density profiles obtained from the two models. We see that convergence between the two models increases with higher α. Fig. 6-26 compares the surface charge distribution around a particle for both models. The two models agree

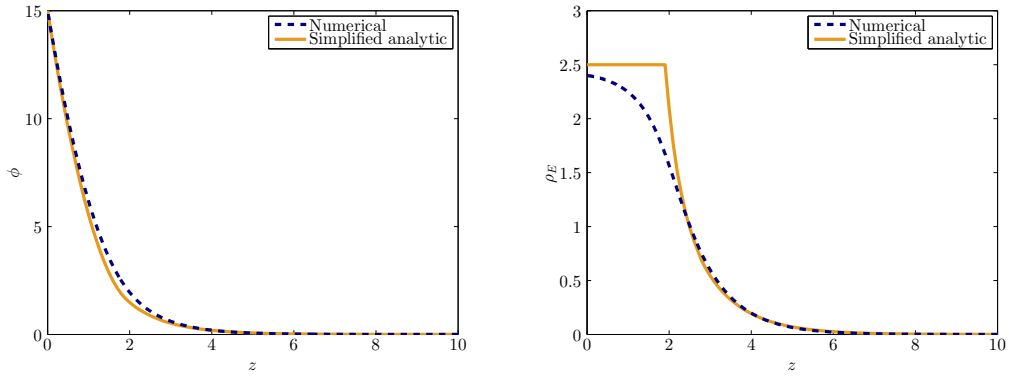

Figure 6-24: Comparison of the electric potential (left) and charge density (right) profiles in the EDL for the numerical and the analytic condensed layer models. The dielectric decrement parameter used in these simulations is $\alpha = 0.1$, the steric parameter used is $\nu = 0.26$, and the zeta potential used is $\zeta = 15$.

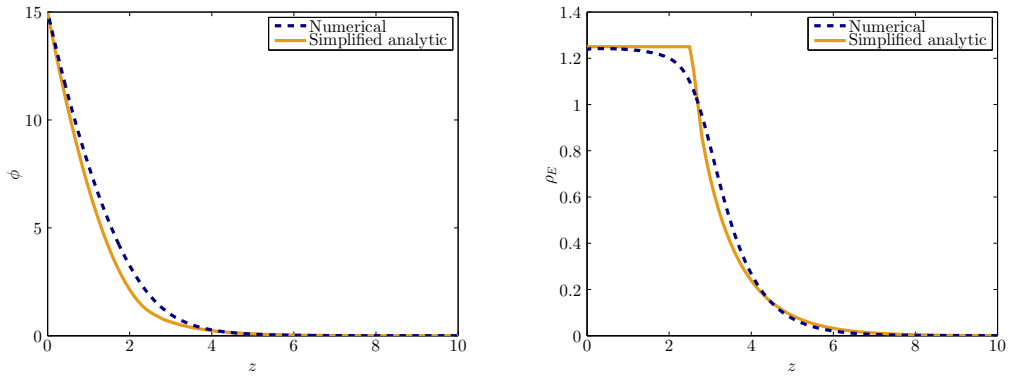

Figure 6-25: Comparison of the electric potential (left) and charge density (right) profiles in the EDL for the numerical and the analytic condensed layer models. The dielectric decrement parameter used in these simulations is $\alpha = 0.2$, the steric parameter used is $\nu = 0.26$, and the zeta potential used is $\zeta = 15$.

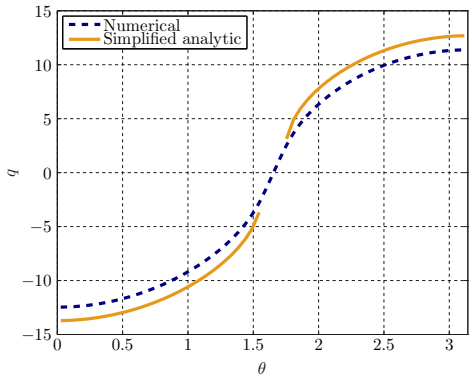

Figure 6-26: Comparison of the surface charge distribution around a positively-charged particle ($Q = 1$) for the numerical and the analytic condensed layer models. The dielectric decrement parameter used in these simulations is $\alpha = 0.1$, and the steric parameter used is $\nu = 0$. The break in the graph occurs becase the analytic model cannot predict a q for low ζ, as mentioned earlier in Chapter 3.

reasonably, although there is some discrepancy due to the differences in the structure of the EDL assumed by the two models, as shown in Fig. 6-24 for $\alpha = 0.1$.

Finally, we compare the simulation results with the asymptotic analytic expressions for the condensed layer model. Fig. 6-27 suggests that the simulation results agree reasonably with the analytic expressions, especially at higher ν and α (in terms of percentage deviation) since it is in these circumstances that the condensed layer approximation is more valid. The asymptotic expression for the steric-only model predicts the simulation results more closely than the expression for the dielectric decrement-only model since we were only able to model the dielectric permittivity in the latter case with a discontinuous function in order to obtain a system of equations that could be solved analytically. In addition, we can observe from Fig. 6-24 that even at a high value of $\alpha = 0.1$, the condensed layer approximation still somewhat overpredicts the total charge in the double layer for the dielectric decrement case.

6.5 Charge-induced thickening

Before we analyze the simulation results for charge-induced thickening, we list here a model simplification possible when the expression (4.1) proposed by Bazant *et*

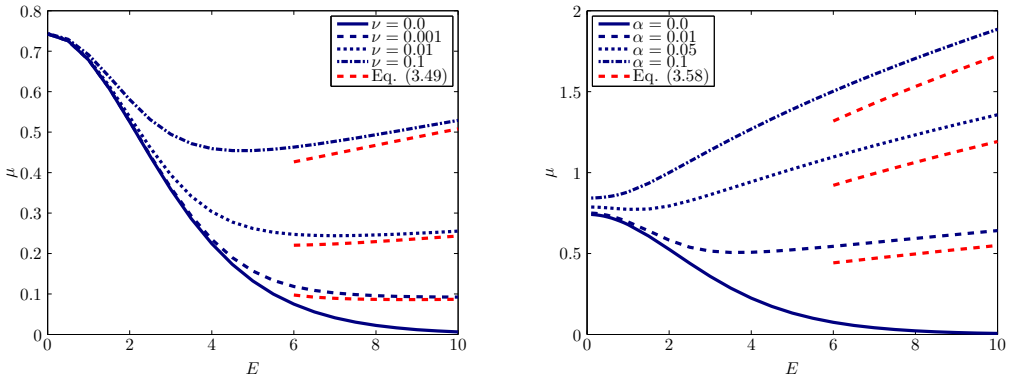

Figure 6-27: Dimensionless electrophoretic mobility of a positively charged particle as a function of the dimensionless applied electric field for different values of the packing parameter ν (left) and of the dielectric decrement parameter α (right). The dimensionless charge of the particle is $Q = 1$. Surface conduction is neglected in these simulations ($\epsilon = 0$). The simulation results in this Figure are compared with the asymptotic analytic expressions (3.49) and (3.58) for the steric-only model and dielectric decrement model respectively.

al. [6] for charge-induced thickening is used. In this case, solving the model using the modified viscosities is equivalent to solving the model without modifying any of the viscosities, but instead using a modified zeta potential

$$\zeta_e = \zeta - \mathrm{sgn}(\zeta) \ln \left(1 + 2\nu \sinh^2 \left(\frac{\zeta}{2} \right) \right). \tag{6.5}$$

In Fig. 6-28, we demonstrate that our simulations generate similar mobilities using the modified viscosity and the modified zeta potential expressions for the charge-induced thickening model proposed by Bazant *et al.* [6]. In addition, we plot the mobilities resulting from the charge-induced thickening models (4.32) and (4.33). At low ν, the different models predict similar mobilities, but at high ν, the different models predict different behavior, especially at low E, suggesting that the choice of the charge-induced thickening model used is relevant to predicting electrophoretic mobilities more accurately. Also, we observe that in general, the charge-induced thickening models predict a depression in the electrophoretic mobility in all cases.

In Fig. 6-29, we observe that charge-induced thickening causes the electrophoretic mobility to saturate at high Q, and even decrease slightly at high Q when surface

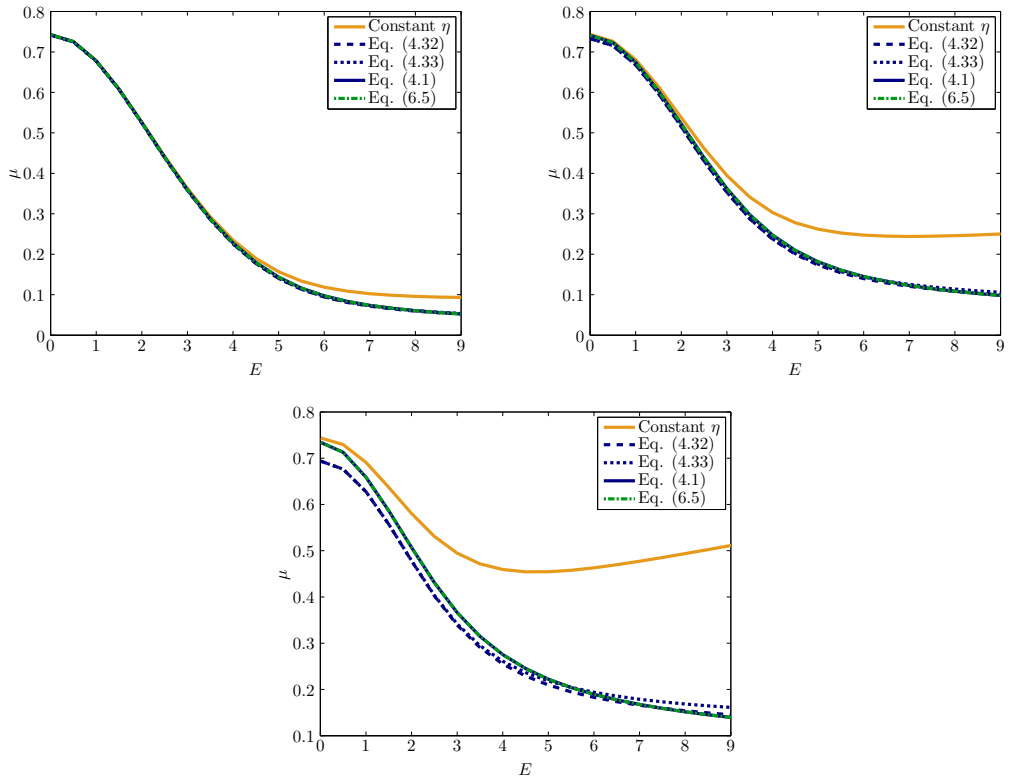

Figure 6-28: Dimensionless electrophoretic mobility of a positively charged particle as a function of the dimensionless applied electric field for different charge-induced thickening models. The dimensionless charge of the particle is $Q = 1$. Surface conduction is neglected in these simulations ($\epsilon = 0$). The parameters for these simulations are $\nu = 0.001$ (upper left), $\nu = 0.01$ (upper right), and $\nu = 0.05$ (bottom).

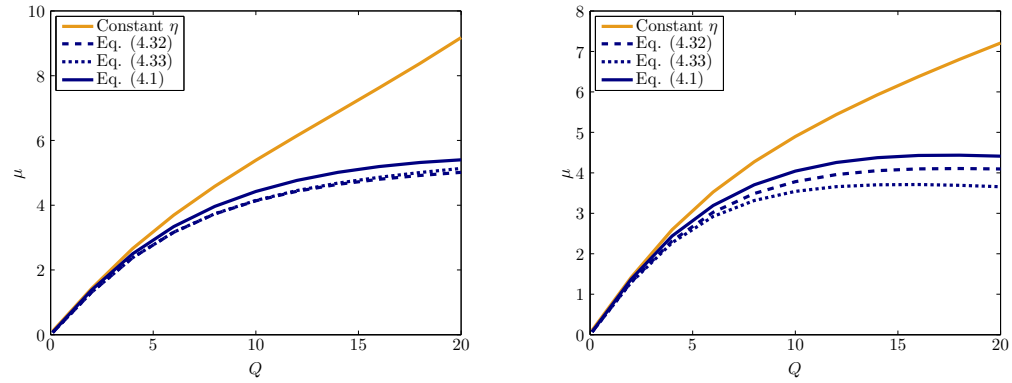

Figure 6-29: Dimensionless electrophoretic mobility of a positively charged particle as a function of the particle charge for different charge-induced thickening models. The values of the ratio ϵ were 0 (left) and 0.02 (right). The dimensionless electric field was held fixed at a value $E = 0.5$, and the packing parameter was held fixed at a value $\nu = 0.05$.

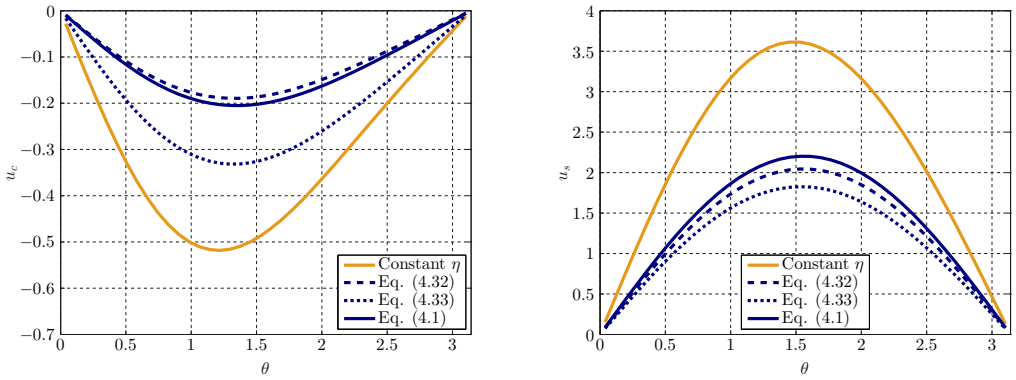

Figure 6-30: Diffusio-osmotic flow (left) and slip velocity (right) at the surface of a charged particle for $Q = 20$, $E = 0.5$, $\nu = 0.05$, $\epsilon = 0.02$ and different charge-induced thickening models.

conduction effects are considered. Also, the two charge-induced thickening models selected in Chapter 4 generate lower mobilities than the model proposed by Bazant *et al.* [6], suggesting that the charge-induced thickening effect may be more significant than previously estimated in some circumstances. In this case, we observe from Fig. 6-30 that charge-induced thickening reduces the magnitude of both the diffusio-osmotic and electro-osmotic slip velocities.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 7

Conclusions

7.1 Findings

In this study, we have presented a numerical model aimed at describing the electrophoresis of ideally polarizable particles. We used the modified Poisson-Boltzmann relation obtained by Kilic *et al.* [4, 5, 61] to account for steric effects due to the finite size of ions in the electric double layer, the further modified Poisson-Boltzmann relation obtained by Hatlo *et al.* [1] to account for dielectric decrement effects in the double layer due to ion-solvent interactions, and various charge-induced thickening models to account for the concentration dependence of viscosity in the double layer and in the bulk solution. In addition, we modeled the ion fluxes in the bulk solution as a combination of electromigration, advection and diffusion. Finally, we incorporated the model developed by Chu and Bazant [3] to match the fields in the double layer and in the bulk solution asymptotically in a manner that accounts for surface conduction. The numerical simulations have demonstrated that this model, when applied to an ideally polarizable particle, yields a non-zero electrophoretic mobility at high electric fields that scales approximately with the square root of the applied electric field when steric effects and dielectric decrement effects are considered, in good agreement with the asymptotic formula obtained by Bazant *et al.* [6] when steric packing is significant. We have also demonstrated that surface conduction influences the electrophoretic mobility. Concentration polarization indeed results in significant

diffusio-osmotic flow around the particle, especially at low fields, which locally reduces the slip velocity. In addition, we showed that at high applied electric fields, advective transport in the bulk solution reduces concentration polarization. We were able to demonstrate that dielectric decrement effects enhance the electrophoretic mobility at high charges, and that the dielectric decrement model reduces to a steric packing model when ion-solvent interactions are small and the particle charge is low. Finally, we showed that charge-induced thickening depresses the electrophoretic mobility, and offered some insights into the impact of asymmetric ionic diffusivities and sizes in the electrolyte.

7.2 Future extensions

The non-linear relation between the applied electric field and the electrophoretic mobility, which in turn is dependent on the bulk concentration and the ionic and particle sizes, could potentially open the way for new separation techniques and deposition methods for ideally polarizable particles. In the case of dielectric particles, Chimenti [126] and later Dukhin *et al.* [127] proposed a scheme for aqueous electrophoretic deposition in asymmetric alternating current (AC) electric fields, which relies on the non-linear regime at high applied voltages. This scheme was successfully tested by Neirinck *et al.* [128] in 2009, and could be adapted to the case of ideally polarizable particles. In doing so, we hope to obtain experimental validation of our simulation results.

In addition, Bazant, Storey and co-workers [129, 130] have recently developed a model aimed at describing the physics of the electric double layer when the mean-field approximation breaks down. This model accounts for subtle electrokinetic effects occurring at large applied voltages. A natural extension of our study could be the development of a numerical model accounting for the breakdown of the mean-field approximation at large applied fields and at high ionic concentrations.

Also, the review by Bazant *et al.* [6] discusses other observations and phenomena like flow reversal, Faradaic charging and ion-surface interactions, in addition to

alternative approximations and approaches like the weighted-density approximation and molecular dynamics simulations. These could be worthwhile extensions to our model to obtain a more physically realistic model of the phenomenon of non-linear electrophoresis of ideally polarizable particles.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A

Code for simulations

A.1 Pseudospectral grid

A.1.1 Generic Chebyshev grid: cheb.m

```
% Source: Spectral Methods in MATLAB, Lloyd N. Trefethen
%
% cheb computes the differentiation matrix D and the Chebyshev grid x in
% 1 dimension for N points
% ------------------------------------------------------------------------

  function [D,x] = cheb(N)
  if N==0, D=0; x=1; return, end
  x = cos(pi*(0:N)/N)';
  c = [2; ones(N-1,1); 2].*(-1).^(0:N)';
  X = repmat(x,1,N+1);
  dX = X-X';
  D  = (c*(1./c)')./(dX+(eye(N+1)));      % off-diagonal entries
  D  = D - diag(sum(D'));                 % diagonal entries
```

A.2 Differentiation operators

A.2.1 Chebyshev grid for radial coordinate r: DM_TL.m

```
% Source: Kevin Chu's PhD thesis / 2005 (Dept of Mathematics, MIT)
%
% DM_TL computes the differentiation matrix and grid points for a rational
% Chebyshev function basis expansion of order N with scale factor L
%
% Usage: function [D, x] = DM_TL (N, L)
%
% Inputs:
% N (required): highest order basis function to include
% L (required): scale factor for transformation between Chebyshev and
%               rational Chebyshev basis
%
% Outputs:
% D: differentiation matrix
% x: grid points
%
%
% NOTES:
% (1) The formula for the derivatives is derived by using the coordinate
% transformation
%
% x = L * (1+y) / (1-y)
%
% where -1 <= y <= 1 is the domain of the Chebyshev basis.
%
% (2) The point x = \infty is NOT excluded from the x-grid points.
%
% (3) This function depends on the CHEB function by Trefethen (2000).
% ------------------------------------------------------------------------------

function [D, x] = DM_TL(N, L)
```

```
32
33 % check inputs
34 if (nargin < 2)
35     error('MATLAB:missingArgs','DM_TL:missing arguments');
36     return
37 end
38
39 % compute differentiation matrix in the y-domain
40 [D_y, y] = cheb(N);
41 one_minus_y = spdiags(1 - y, 0, N + 1, N + 1);
42
43 % transform to the x-domain
44 D = 0.5/L*(one_minus_y^2)*D_y;
45 x = L*(1+y)./(1-y);
```

A.2.2 Periodic grid for polar coordinate θ: DM_cosine_interior.m

```
1  % Source: Kevin Chu's PhD thesis / 2005 (Dept of Mathematics, MIT)
2  %
3  % DM_cosine_interior computes the differentiation matrix and grid points
4  % for a cosine cardinal basis expansion of order N with all grid points in
5  % the interior of the domain
6  %
7  % Usage: function [D, theta, Delta_theta] = DM_cosine_interior (N)
8  %
9  % Input:
10 % N (required): highest Fourier component to include
11 %
12 % Outputs:
13 % D:            differentiation matrix
14 % theta:        grid points
15 % Dtheta:       angular step
16 %
17 % NOTES:
18 % (1) The formula for the derivatives was derived from the formula for the
```

```
19 % cardinal functions given in Boyd (2000) Appendix F.5
20 % ------------------------------------------------------------------------
21
22 function [D, theta, Dtheta] = DM_cosine_interior (N)
23
24 % N=0 case
25 if N == 0
26     D = 0;
27     theta = 0;
28     return
29 end
30
31 % N>0 case
32 j = 1:N;
33 theta = (2*j' - 1) * pi / 2 / N;
34 c = ones(1,N) .* (-1).^( (1:N) + 1 );
35 T = repmat(theta,1,N);
36
37 % off-diagonal entries
38 %
39 % NOTES:
40 % (1) The eye(N) avoids division by zero on the diagonal, which does not
41 % contain essential elements
42 % (2) The sin(N*T) term just gives an alternating sequence of 1's and -1's
43 % down a column
44
45 off_diag_D = repmat(c,N,1) .* sin(N*T) .* sin(T') ./ ...
46     (cos(T') - cos(T) + eye(N));
47
48 % diagonal entries
49 diag_D = -0.5*cot(theta);
50
51 % assemble matrix
52 D = triu(off_diag_D, 1) + tril(off_diag_D, -1) + diag(diag_D);
53
54 % Angular discretization step
```

```
Dtheta = (1:N)* pi/ N - (0:N - 1)* pi/ N;
```

A.2.3 Divergence operator: div.m

```
% Source: Kevin Chu's PhD thesis / 2005 (Dept of Mathematics, MIT)
%
% div computes the discrete divergence operator in spherical coordinates
% with azimuthal symmetry
%
% Usage: function D_div = div(D_r, D_theta, r, theta)
%
% Inputs:
% D_r (required):       differentiation matrix in radial direction
% D_theta (required):   differentiation matrix in polar angle direction
% r (required):         radial grid points
% theta (required):     polar angle grid points
%
% Output:
% D_div: Discrete divergence operator for solutions stored in radial-major
%    order (i.e. as one goes down the sequence, r decreases from infinity,
%    then theta increases by one step, and r decreases from infinity again).
%    The components of D_div are the operators for the radial and polar
%    angle components of the vector field.
% ------------------------------------------------------------------------

function D_div = div(D_r, D_theta, r, theta)

num_gridpts_r = length(r);
num_gridpts_theta = length(theta);

one_over_r = spdiags(1./r, 0, num_gridpts_r, num_gridpts_r);
sin_theta = spdiags(sin(theta), 0, num_gridpts_theta, num_gridpts_theta);
one_over_sin_theta = spdiags(1./sin(theta), 0, ...
    num_gridpts_theta, num_gridpts_theta);

```

```
D_div = {kron(speye(num_gridpts_theta), 2*one_over_r + D_r), ...
    kron(one_over_sin_theta*D_theta*sin_theta, one_over_r)};
```

A.2.4 Divergence operator on particle surface: div_s.m

```
% Source: Kevin Chu's PhD thesis / 2005 (Dept of Mathematics, MIT)
%
% div_s computes the discrete surface divergence operator in spherical
% coordinates with azimuthal symmetry
%
% Usage: function D_div_s = div_s(D_theta, theta, r)
%
% Inputs:
% D_theta (required):   differentiation matrix in polar angle direction
% theta (required):     polar angle grid points
% r (required):         radius of sphere
%
% Output:
% D_div_s:  Discrete surface divergence operator for problems with azimuthal
%       symmetry
% ------------------------------------------------------------------------

function D_div_s = div_s(D_theta, theta, r)

num_gridpts_theta = length(theta);

sin_theta_vec = sin(theta);
sin_theta = spdiags(sin_theta_vec, 0, num_gridpts_theta, num_gridpts_theta);
one_over_sin_theta = spdiags(1./sin_theta_vec, 0, ...
    num_gridpts_theta, num_gridpts_theta);

D_div_s = one_over_sin_theta*D_theta*sin_theta/r;
```

A.2.5 Gradient operator: grad.m

```
% Source: Kevin Chu's PhD thesis / 2005 (Dept of Mathematics, MIT)
%
% grad computes the discrete gradient operator in spherical coordinates
% with azimuthal symmetry
%
% Usage: function D_grad = grad(D_r, D_theta, r, theta)
%
% Inputs:
% D_r (required):       differentiation matrix in radial direction
% D_theta (required):   differentiation matrix in polar angle direction
% r (required):         radial grid points
% theta (required):     polar angle grid points
%
% Output:
% D_grad: Discrete gradient operator for solutions stored in radial—major
% order. The components of grad are the operators for the radial and polar
% angle components of the vector field.
%
% NOTES:
% (1) This code uses sparse matrices for identity and diagonal matrices to
% save memory and improve performance.
% ————————————————————————————————————————————————————————————————————————

function D_grad = grad(D_r, D_theta, r, theta)

num_gridpts_r = length(r);
num_gridpts_theta = length(theta);

one_over_r = spdiags(1./r, 0, num_gridpts_r, num_gridpts_r);
D_grad = {kron(speye(num_gridpts_theta), D_r), kron(D_theta, one_over_r)};
```

A.2.6 Laplacian operator: laplacian.m

```
% Source: Kevin Chu's PhD thesis / 2005 (Dept of Mathematics, MIT)
%
```

```
% laplacian computes the discrete Laplacian operator in spherical
% coordinates with azimuthal symmetry
%
% Usage: function D_laplacian = laplacian(D_r, D_theta, r, theta)
%
% Inputs:
% D_r (required):       differentiation matrix in radial direction
% D_theta (required):   differentiation matrix in polar angle direction
% r (required):         radial grid points
% theta (required):     polar angle grid points
%
% Output:
% D_laplacian: Discrete Laplacian operator for solutions stored in
% radial-major order.
% ------------------------------------------------------------------------

function D_laplacian = laplacian(D_r, D_theta, r, theta)

num_gridpts_r = length(r);
num_gridpts_theta = length(theta);

one_over_r = spdiags(1./r, 0, num_gridpts_r, num_gridpts_r);
sin_theta = spdiags(sin(theta), 0, num_gridpts_theta, num_gridpts_theta);
one_over_sin_theta = spdiags(1./sin(theta), 0, ...
    num_gridpts_theta, num_gridpts_theta);

D_laplacian = kron(speye(num_gridpts_theta), 2*one_over_r*D_r + D_r^2) + ...
    kron(one_over_sin_theta*D_theta*sin_theta*D_theta, one_over_r^2);
```

A.2.7 Operators required for solving Stokes equation: stokes.m

```
% stokes computes the discrete Stokes operator in spherical
% coordinates with azimuthal symmetry
%
% Usage: [S, P] = stokes(G, D_r, D_theta, r, theta)
```

```
%
% Inputs:
% G (required):         discrete gradient operator
% D_r (required):       differentiation matrix in radial direction
% D_theta (required):   differentiation matrix in polar angle direction
% r (required):         radial grid points
% theta (required):     polar angle grid points
%
% Outputs:
% S:    Discrete Stokes operator for solutions stored in
%   radial-major order.
% P:    Potential operator
% ------------------------------------------------------------------

function [S, P] = stokes(G, D_r, D_theta, r, theta)

num_gridpts_r = length(r);
num_gridpts_theta = length(theta);

% ------------------------------------
% Step 1: Define some cache common expressions
% ------------------------------------

sin_theta = spdiags(sin(theta), 0, num_gridpts_theta, num_gridpts_theta);
cot_theta = spdiags(cot(theta), 0, num_gridpts_theta, num_gridpts_theta);
cos_theta = spdiags(cos(theta), 0, num_gridpts_theta, num_gridpts_theta);

sin_theta_square = spdiags(sin(theta).*sin(theta), 0, ...
    num_gridpts_theta, num_gridpts_theta);
one_over_sin_theta = spdiags(1./sin(theta), 0, ...
    num_gridpts_theta, num_gridpts_theta);

r_full = spdiags(r, 0, num_gridpts_r, num_gridpts_r);
r_square = spdiags(r.^2, 0, num_gridpts_r, num_gridpts_r);
one_over_r = spdiags(1./r, 0, num_gridpts_r, num_gridpts_r);

```

```
% ------------------------------------
% Step 2: Define the operators
% ------------------------------------

% Vorticity operator (modified to accommodate our modified stream function)
L = - 0.5 * kron(sin_theta_square, r_square) * ...
    ( kron( speye(num_gridpts_theta), 4*one_over_r*D_r + D_r^2 ) + ...
    kron( 3.0*cot_theta*D_theta + D_theta^2, one_over_r^2) );

% Helper operator to use in Stokes operator
E = G{1}*G{1} + kron(sin_theta, speye(num_gridpts_r) )*G{2} ...
    *kron(one_over_sin_theta, speye(num_gridpts_r) )*G{2};

% Stokes operator (to operate on stream function in Stokes equation)
S = kron(one_over_sin_theta, one_over_r) * E * L;

% Velocity field operator (to extract velocities from stream function)
P = {0.5 * kron(sin_theta*D_theta, speye(num_gridpts_r)) + ...
    kron(cos_theta, speye(num_gridpts_r)), ...
    - 0.5 * kron(sin_theta, r_full*D_r) ...
    - kron(sin_theta, speye(num_gridpts_r))};
```

A.3 Other helper functions (Steric model)

A.3.1 Calculate diffusio-osmotic velocity: chemiphoresis.m

```
% chemiphoresis computes the diffusio-osmotic velocity at the surface of
% the particle.
%
% function [u_cp, Du_cp_Dzeta] = chemiphoresis(xi, zeta, nu, res_tol)
%
% Inputs:
% xi (required):        Lower limit for integration
% zeta (required):      Zeta potential at the surface of the particle
```

```
% nu (required):        Steric parameter
% res_tol (optional):   Tolerance for residual in computation of u_cp
%
% Outputs:
% u_cp:          Diffusio-osmotic velocity
% Du_cp_Dzeta:  Derivative of the diffusio-osmotic velocity with respect to
%   the zeta potential
% ------------------------------------------------------------------------

function [u_cp, Du_cp_Dzeta] = chemiphoresis(xi, zeta, nu, res_tol)

% ---------------------------------------------------------------
% STEP 1: check argument list and set default values
% ---------------------------------------------------------------

max_args = 4;
if (nargin < 3)
    error('MATLAB:missingArgs','chemiphoresis:missing arguments');
end
if (nargin < 4)
    res_tol = 0.01;
end

% ---------------------------------------------------------------
% STEP 2: Initialization
% ---------------------------------------------------------------
num_gridpts = length(zeta);

% ---------------------------------------------------------------
% STEP 3: Sort the values of the zeta potential by ascending order
% ---------------------------------------------------------------

signZeta = sign(zeta);
[zeta_s, idx] = sort(abs(zeta));

% ---------------------------------------------------------------
```

```
% STEP 4: Perform the double integration
% ------------------------------------------------------------

u_cp = spalloc(num_gridpts, 1, 0);
Du_cp_Dzeta = spalloc(num_gridpts, 1, 0);

v_min = xi + eps;
res_w = 0.0;

for n = 1:num_gridpts

    % Initialization
    v_max = zeta_s(n);

    % Evaluate the integral
    if (v_min < 1E-5)
        res_w = res_w + dblquad(@dbfun_v_small, 0, 1, v_min, 1E-5, res_tol) +...
            dblquad(@dbfun_u_small, 0, 1E-5, 1E-5, v_max, res_tol) +...
            dblquad(@dbfun, 1E-5, 1, 1E-5, v_max, res_tol);
    else
        res_w = res_w +...
            dblquad(@dbfun_u_small, 0, 1E-5, v_min, v_max, res_tol) +...
            dblquad(@dbfun, 1E-5, 1, v_min, v_max, res_tol);
    end

    % Derivative w.r.t the zeta potential
    fun = @(u) dbfun(u, v_max);
    fun_u_small = @(u) dbfun_u_small(u,v_max);
    Du_cp_Dzeta(n) = quad(fun_u_small, 0, 1E-5, res_tol) +...
        quad(fun, 1E-5, 1, res_tol);

    % Update
    u_cp(n) = res_w;
    v_min = v_max;

end
```

```
81
82 % ------------------------------------------------------------
83 % STEP 5: Re-sort the array values
84 % ------------------------------------------------------------
85
86 u_cp(idx) = u_cp;
87 Du_cp_Dzeta(idx) = Du_cp_Dzeta;
88
89
90 % ------------------------------------------------------------
91 % Sub-functions: Define the functions to integrate
92 % ------------------------------------------------------------
93
94 % Function split into cases as MATLAB cannot handle the function well
95 % at small v
96
97 function result = dbfun(u,v)
98 if (nu == 0)
99     result = 0.5 .* v .* sinh(0.5 .* u .* v)./sinh(0.5 * v);
100 else
101     result = 0.5 .* v .* sqrt( log(1 + 2 * nu * sinh(0.5 .* u .* v).^2)./ ...
102         log(1 + 2 * nu * sinh(0.5 * v).^2));
103 end
104 end
105
106 function result = dbfun_u_small(u,v)
107 if (nu == 0)
108     result = 0.5 * v * sinh(0.5 .* u .* v)./sinh(0.5 .* v);
109 else
110     result = 0.5 * v * sqrt(2*nu) .* sinh(0.5 .* u .* v) ./ ...
111         sqrt(log(1 + 2 * nu * sinh(0.5 * v).^2));
112 end
113 end
114
115 function result = dbfun_v_small(u,v)
116 result = 0.5 * v .* sinh(0.5 .* u .* v) ./ sinh(0.5 .* v);
```

```
117 end
118
119 end
```

A.3.2 Solve for zeta potential distribution around particle: computeZetaPotential.m

```
1  % Adapted from Kevin Chu's PhD thesis / 2005 (Dept of Mathematics, MIT)
2  %
3  % computeZetaPotential computes the zeta potential on the surface of the
4  % particle using the charge conservation of the particle.
5  %
6  %function [V, zeta, q, w, Dzeta_Dpsi, Dzeta_Dc_s, Dq_Dpsi, Dq_Dc_s, ...
7  %    Dw_Dpsi, Dw_Dc_s, p1, p2, p3, p4, Dp1_Dpsi, Dp1_Dc_s, Dp2_Dpsi, ...
8  %    Dp2_Dc_s, Dp3_Dpsi, Dp3_Dc_s, Dp4_Dpsi, Dp4_Dc_s ...
9  %    ] = computeZetaPotential( ...
10 %    Psi, c_s, nu, Q, V_init, ... % physical parameters
11 %    res_tol, delta_zeta_tol, max_iters, ... % iteration parameters
12 %    show_stats ... % show statistics flag
13 %   )
14 %
15 % Inputs:
16 % Psi (required):              Surface potential
17 % c_s (required):              Surface concentration
18 % nu (required):               Packing parameter
19 % Q (required):                Total surface charge of the particle
20 % V_init (optional):           Initial guess for the potential of the particle
21 % res_tol (optional):          Tolerance for residual in computation of zeta
22 % delta_zeta_tol (optional):   Tolerance for change in zeta
23 % max_iters (optional):        Maximum number of Newton iterations
24 % show_stats (optional):       Show statistics if set to a non-zero value
25 %
26 % Outputs:
27 % V:            Potential of the particle
```

```
% zeta:          Zeta potential at each point of the surface
% q:             Excess charge
% w:             Excess concentration
% Dzeta_Dpsi:    Derivative of zeta with respect to the bulk potential
% Dzeta_Dc_s:    Derivative of zeta with respect to the surface concentration
% Dq_Dpsi:       Derivative of q with respect to the bulk potential
% Dq_Dc_s:       Derivative of q with respect to the surface concentration
% Dw_Dpsi:       Derivative of w with respect to the bulk potential
% Dw_Dc_s:       Derivative of w with respect to the surface concentration
% Note: p_i are additional quantities to account for advection, and their
% deriviatives with respect to the bulk potential and surface concentration
% are defined accordingly
% ------------------------------------------------------------------------

function [V, zeta, q, w, Dzeta_Dpsi, Dzeta_Dc_s, Dq_Dpsi, Dq_Dc_s, ...
    Dw_Dpsi, Dw_Dc_s, p1, p2, p3, p4, Dp1_Dpsi, Dp1_Dc_s, Dp2_Dpsi, ...
    Dp2_Dc_s, Dp3_Dpsi, Dp3_Dc_s, Dp4_Dpsi, Dp4_Dc_s ...
    ] = computeZetaPotential( ...
    Psi, c_s, nu, Q, V_init, ... % physical parameters
    res_tol, delta_zeta_tol, max_iters, ... % iteration parameters
    show_stats ... % show statistics flag
    )

% ------------------------------------------------------------
% STEP 1: check argument list and set default values
% ------------------------------------------------------------

max_args = 9;
if (nargin < 4)
    error('MATLAB:missingArgs','computeZetaPotential:missing arguments');
end
if (nargin < max_args)
    show_stats = 0;
end
if (nargin < max_args - 1)
    max_iters = 100;
```

```
64 end
65 if (nargin < max_args - 2)
66     delta_zeta_tol = 1E-13;
67 end
68 if (nargin < max_args - 3)
69     res_tol = 1E-8;
70 end
71 if (nargin < max_args - 4)
72     V_init = 0.0;
73 end
74
75 % ------------------------------------------------------------------
76 % STEP 2: Define the integration operator
77 % ------------------------------------------------------------------
78
79 N_theta = length(Psi);
80 [~, theta, DeltaTheta] = DM_cosine_interior(N_theta);
81
82 % ------------------------------------------------------------------
83 % STEP 3: Compute the initial residual
84 % ------------------------------------------------------------------
85
86 % Initialize iteration
87 V = V_init;
88 delta_V = 1;
89 res = 1;
90 norm_res = norm(res, inf);
91 norm_delta_zeta = norm(delta_V, inf);
92 count = 0;
93
94 % Calculate the initial residual
95 if (nu == 0)
96    res = Q - DeltaTheta * 2.0 *  ( sqrt(c_s) .* sinh( 0.5*(V - Psi) ) ...
97        .* sin(theta) );
98 else
99    res = Q - DeltaTheta * ( sign(V - Psi) .* sqrt( (2.0/nu) * c_s .* ...
```

```
100         log(1 + 2*nu*sinh(0.5*(V - Psi)).^2 ) ) .* sin(theta) );
101 end
102
103 % ------------------------------------------------------------
104 % STEP 4: Newton iteration
105 % ------------------------------------------------------------
106
107 while (norm_res > res_tol && norm_delta_zeta > delta_zeta_tol && count < max_iters)
108
109     if (nu == 0)
110         J = - 2.0 * DeltaTheta * ( 0.5 * sqrt(c_s) .* cosh( 0.5*(V - Psi) ) ...
111             .* sin(theta) );
112     else
113     J = - DeltaTheta * (...
114     sign(V - Psi) .* sqrt(nu * c_s/2.0) .* sinh(V - Psi) .* sin(theta) ./ ...
115     ( ( 1 + 2.0 * nu * sinh(0.5*(V - Psi)).^2 ).* ...
116     sqrt( log(1 + 2*nu*sinh(0.5*(V - Psi)).^2 ) ) ) ...
117     );
118     end
119
120     delta_V = - res/J;
121     V = V + delta_V;
122
123     if (nu == 0)
124     res = Q - 2.0 * DeltaTheta * ( sqrt(c_s) .* sinh( 0.5*(V - Psi) ) ...
125         .* sin(theta) );
126     else
127     res = Q - DeltaTheta * ( sign(V - Psi) .* sqrt( (2.0/nu) * c_s .* ...
128         log(1 + 2*nu*sinh(0.5*(V - Psi)).^2 ) ).* sin(theta) );
129     end
130
131     % Update norm_res, norm_delta_zeta and count
132     norm_res = norm(res, inf);
133     norm_delta_zeta = norm(delta_V, inf);
134     count = count + 1;
135
```

```
136 end
137
138 % ------------------------------------------------------------
139 % STEP 5: Output
140 % ------------------------------------------------------------
141
142 zeta = V - Psi;
143
144 if(nu == 0)
145
146     q = - 4.0 * sqrt(c_s) .* sinh(zeta/2);
147     w = 4.0 * sqrt(c_s) .* (sinh(zeta/4)).^2 * 2;
148
149     Dzeta_Dpsi = 0.0;
150     Dzeta_Dc_s = -tanh(zeta/2)./c_s;
151
152     Dq_Dc_s = 0.5 * q./c_s - 2.0 * sqrt(c_s) .* cosh (zeta/2) .* Dzeta_Dc_s;
153     Dq_Dpsi = - 2.0 * sqrt(c_s) .* cosh (zeta/2) .* Dzeta_Dpsi;
154
155     Dw_Dc_s = 0.5 * w ./ c_s - 0.25 * q .* Dzeta_Dc_s ;
156     Dw_Dpsi = -0.25 * q.*Dzeta_Dpsi;
157
158 else
159     %# Zeta potential
160
161     Dzeta_Dpsi = 0.0;
162     Dzeta_Dc_s = -log(1.0 + 2.0 * nu * sinh(0.5*zeta).^2) .* ...
163         (1.0 + 2.0 * nu * sinh(0.5*zeta).^2) ./ (nu * c_s .* sinh(zeta)) ;
164
165     %# Excess Charge
166
167     q = - 2.0 * sign(zeta) .* sqrt( (2.0/nu) * c_s .* ...
168         log(1 + 2*nu*sinh(0.5*(V - Psi)).^2 ) );
169     Dq_Dzeta = - sign(zeta) .* sqrt(2.0* nu *c_s) .* sinh(zeta) ./ ...
170         ( ( 1 + 2.0 * nu * sinh(0.5*(zeta)).^2 ) .* ...
171         sqrt( log(1 + 2*nu*sinh(0.5*(zeta)).^2 ) ) );
```

```
172     Dq_Dc_s = 0.5 * q./c_s + Dq_Dzeta .* Dzeta_Dc_s;
173     Dq_Dpsi = Dq_Dzeta .* Dzeta_Dpsi;
174
175     %# Excess Concentration
176
177     [w, Dw_Dzeta] = dukhin(zeta, c_s, nu, res_tol);
178     w = w .* sqrt(c_s) * 2;
179     Dw_Dc_s = sqrt(c_s).*Dw_Dzeta .* Dzeta_Dc_s .* sign(zeta) + 0.5*w./c_s;
180     Dw_Dpsi = sqrt(c_s).*Dw_Dzeta.*Dzeta_Dpsi;
181
182 end
183
184 [p1, Dp1_Dzeta] = peclet1(zeta, c_s, nu, res_tol);
185 p1 = p1 .* sqrt(c_s) * 2;
186 Dp1_Dc_s = sqrt(c_s).*Dp1_Dzeta .* Dzeta_Dc_s .* sign(zeta) + 0.5*p1./c_s;
187 Dp1_Dpsi = sqrt(c_s).*Dp1_Dzeta.*Dzeta_Dpsi;
188
189 [p2, Dp2_Dzeta] = peclet2(zeta, c_s, nu, res_tol);
190 p2 = p2 .* sqrt(c_s) * 2;
191 Dp2_Dc_s = sqrt(c_s).*Dp2_Dzeta .* Dzeta_Dc_s .* sign(zeta) + 0.5*p2./c_s;
192 Dp2_Dpsi = sqrt(c_s).*Dp2_Dzeta.*Dzeta_Dpsi;
193
194 [p3, Dp3_Dzeta] = peclet3(zeta, c_s, nu, res_tol);
195 p3 = p3 .* sqrt(c_s) * 2;
196 Dp3_Dc_s = sqrt(c_s).*Dp3_Dzeta .* Dzeta_Dc_s .* sign(zeta) + 0.5*p3./c_s;
197 Dp3_Dpsi = sqrt(c_s).*Dp3_Dzeta.*Dzeta_Dpsi;
198
199 [p4, Dp4_Dzeta] = peclet4(zeta, c_s, nu, res_tol);
200 p4 = p4 .* sqrt(c_s) * 2;
201 Dp4_Dc_s = sqrt(c_s).*Dp4_Dzeta .* Dzeta_Dc_s .* sign(zeta) + 0.5*p4./c_s;
202 Dp4_Dpsi = sqrt(c_s).*Dp4_Dzeta.*Dzeta_Dpsi;
203
204 % ------------------------------------------------------------
205 % STEP 6: Error checking
206 % ------------------------------------------------------------
207
```

```
208 % Throw a warning if the solution has not converged
209 if (norm_res > res_tol && norm_delta_zeta > delta_zeta_tol)
210     mesg_id = 'computeZetaPotential:solutionNotConverged';
211     warning(mesg_id, ...
212         ...%String truncated in thesis for brevity
213     'cZP: Solution NOT converged! ...res = %0.10f, delta_zeta = %0.10f', ...
214     norm_res, norm_delta_zeta);
215 end
216
217 % ------------------------------------------------------------
218 % STEP 7: Show computation statistics if requested
219 % ------------------------------------------------------------
220 if (show_stats)
221     stats_string = sprintf( ...
222         ...%String truncated in thesis for brevity
223         '\n cZP \n Res = %0.5g\n # iters = %d\n Change in sol = %0.5g\n\n', ...
224         norm_res, count, norm_delta_zeta);
225     disp(stats_string);
226 end
```

A.3.3 Calculate excess charge: dukhin.m

```
1  % dukhin computes the excess charge number on the surface knowing the zeta
2  % potential of the particle.
3  %
4  % function [w, Dw_Dzeta] = dukhin(zeta, c_s, nu, res_tol)
5  %
6  % Inputs:
7  % zeta (required):      Zeta potential at the surface of the particle
8  % c_s (required):       Ionic concentration in the bulk solution at the
9  %       boundary of the particle
10 % nu (required):        Steric parameter
11 % res_tol (optional):   Tolerance for residual in computation of w
12 %
13 % Output:
```

```
14 % w:            Excess concentration
15 % Dw_Dzeta:     Derivative of w with respect to the zeta potential
16 % ------------------------------------------------------------------------------
17
18 function [w, Dw_Dzeta] = dukhin(zeta, c_s, nu, res_tol)
19
20 % ------------------------------------------------------------
21 % STEP 1: check argument list and set default values
22 % ------------------------------------------------------------
23
24 if (nargin < 1)
25     error('MATLAB:missingArgs','dukhin:missing arguments');
26 end
27 if (nargin < 4)
28     res_tol = 0.01;
29 end
30
31 % ------------------------------------------------------------
32 % STEP 2: Initialization
33 % ------------------------------------------------------------
34 num_gridpts = length(zeta);
35
36 % ------------------------------------------------------------
37 % STEP 3: Sort the values of the zeta potential by ascending order
38 % ------------------------------------------------------------
39
40 [zeta_s, idx] = sort(abs(zeta));
41
42 % ------------------------------------------------------------
43 % STEP 4: Define the function to integrate
44 % ------------------------------------------------------------
45
46 fun_small = @(u) ( (abs(u)/2.0)*(1.0 - nu)./( (1.0 + 1.0/2.0*nu*u.^2)));
47 fun = @(u)  ( (cosh(u) - 1.0)*(1.0 - nu) )./( (1.0 + 2.0*nu*sinh(0.5*u).^2) ...
48     .*sqrt((2.0/nu).*log(1.0 + 2.0*nu*sinh(0.5*u).^2)) );
49
```

```
% ------------------------------------------------------------
% STEP 5: Perform the integration
% ------------------------------------------------------------

w = spalloc(num_gridpts, 1, 0);
Dw_Dzeta = spalloc(num_gridpts, 1, 0);

x_min = 0.0;
res_w = 0.0;

for n = 1:num_gridpts

    % Initialization
    x_max = zeta_s(n);

    % Evaluate the integral
    if (x_min < 1E-5 && x_max < 1E-5)
        res_w = res_w + quad(fun_small, x_min, x_max, res_tol);
    elseif (x_min < 1E-5 && x_max > 1E-5)
        res_w = res_w + quad(fun_small, x_min, 1E-5, res_tol) + ...
            quad(fun, 1E-5, x_max, res_tol);
    else
        res_w = res_w + quad(fun, x_min, x_max, res_tol);
    end

    % Update
    w(n) = res_w;
    Dw_Dzeta(n) = fun(x_max);
    x_min = x_max;

end

% ------------------------------------------------------------
% STEP 6: Re-sort the array values
% ------------------------------------------------------------

```

```
86 w(idx) = w;
87 Dw_Dzeta(idx) = Dw_Dzeta;
```

A.3.4 Calculate advection quantity p_1: peclet1.m

```
1  % peclet1 computes the quantity p_1 required to account for advection in
2  % the excess surface flux around the particle.
3  %
4  % function [p1, Dp1_Dzeta] = peclet1(zeta, c_s, nu, res_tol)
5  %
6  % Inputs:
7  % zeta (required):      Zeta potential at the surface of the particle
8  % c_s (required):       Bulk concentration
9  % nu (required):        Steric parameter
10 % res_tol (optional):   Tolerance for residual in computation of w
11 %
12 % Output:
13 % p1:           Desired quantity
14 % Dp1_Dzeta:    Derivative of p1 with respect to the zeta potential
15 % ------------------------------------------------------------------------
16
17 function [p1, Dp1_Dzeta] = peclet1(zeta, c_s, nu, res_tol)
18
19 % ---------------------------------------------------------
20 % STEP 1: check argument list and set default values
21 % ---------------------------------------------------------
22
23 max_args = 4;
24 if (nargin < 3)
25     error('MATLAB:missingArgs','peclet1:missing arguments');
26 end
27 if (nargin < 4)
28     res_tol = 0.01;
29 end
30
```

```
31 % ------------------------------------------------------------------
32 % STEP 2: Initialization
33 % ------------------------------------------------------------------
34 num_gridpts = length(zeta);
35
36 % ------------------------------------------------------------------
37 % STEP 3: Sort the values of the zeta potential by ascending order
38 % ------------------------------------------------------------------
39
40 [zeta_s, idx] = sort(abs(zeta));
41
42 % ------------------------------------------------------------------
43 % STEP 4: Perform the integration
44 % ------------------------------------------------------------------
45
46 p1 = spalloc(num_gridpts, 1, 0);
47 Dp1_Dzeta = spalloc(num_gridpts, 1, 0);
48
49 x_min = 0.0;
50 res_w = 0.0;
51
52 for n = 1:num_gridpts
53
54     % Initialization
55     x_max = zeta_s(n);
56     [chem_all,chem_all2] = chemiphoresis(0,x_max,nu);
57
58     fun_small = @(u) fun_small_hold(u,x_max,chem_all);
59     fun = @(u) fun_hold(u,x_max,chem_all);
60     fun_nosteric = @(u) fun_nosteric_hold(u,x_max,chem_all);
61     fun_small2 = @(u) fun_small_hold2(u,x_max,chem_all2);
62     fun2 = @(u) fun_hold2(u,x_max,chem_all2);
63     fun_nosteric2 = @(u) fun_nosteric_hold2(u,x_max,chem_all2);
64
65     % Evaluate the integral
66     if (nu == 0)
```

```
67          res_w = res_w + quad(fun_nosteric, x_min, x_max, res_tol);
68      else
69          if (x_min < 1E-5 && x_max < 1E-5)
70              res_w = res_w + quad(fun_small, x_min, x_max, res_tol);
71          elseif (x_min < 1E-5 && x_max > 1E-5)
72              res_w = res_w + quad(fun_small, x_min, 1E-5, res_tol) + ...
73                  quad(fun, 1E-5, x_max, res_tol);
74          else
75              res_w = res_w + quad(fun, x_min, x_max, res_tol);
76          end
77      end
78
79      % Update
80      p1(n) = res_w;
81      if (nu == 0)
82          Dp1_Dzeta(n) = fun_nosteric(x_max) + ...
83              quad(fun_nosteric2, x_min, x_max, res_tol);
84      else
85          if (x_min < 1E-5 && x_max < 1E-5)
86              Dp1_Dzeta(n) = fun(x_max) + ...
87                  quad(fun_small2, x_min, x_max, res_tol);
88          elseif (x_min < 1E-5 && x_max > 1E-5)
89              Dp1_Dzeta(n) = fun(x_max) + ...
90                  quad(fun_small2, x_min, 1E-5, res_tol) + ...
91                  quad(fun2, 1E-5, x_max, res_tol);
92          else
93              Dp1_Dzeta(n) = fun(x_max) + quad(fun2, x_min, x_max, res_tol);
94          end
95      end
96      x_min = x_max;
97
98  end
99
100 % ------------------------------------------------------------
101 % STEP 5: Re-sort the arrays values
102 % ------------------------------------------------------------
```

```
103
104 p1(idx) = p1;
105 Dp1_Dzeta(idx) = Dp1_Dzeta;
106 Dp1_Dzeta = Dp1_Dzeta.*sign(zeta);
107
108
109 % ------------------------------------------------------------
110 % Functions to integrate for p1
111 % ------------------------------------------------------------
112
113 function result = fun_nosteric_hold(u,max,chem_all)
114 chem_hold = 0;
115 for i = 1:size(u)
116     chem_hold(i) = chemiphoresis(u(i),max,nu);
117 end
118 result = 1/2./sinh(0.5*u) .* (cosh(u).*chem_hold - chem_all);
119 end
120
121 function result = fun_small_hold(u,max,chem_all)
122 chem_hold = 0;
123 for i = 1:size(u)
124     chem_hold(i) = chemiphoresis(u(i),max,nu);
125 end
126 result = 1./abs(u).*(1./(1.0 + 1.0/2.0*nu*u.^2).*chem_hold - chem_all);
127 end
128
129 function result = fun_hold(u,max,chem_all)
130 chem_hold = 0;
131 for i = 1:size(u)
132     chem_hold(i) = chemiphoresis(u(i),max,nu);
133 end
134 result = 1./(sqrt((2.0/nu).*log(1.0 + 2.0*nu*sinh(0.5*u).^2))) .* ...
135     (cosh(u)./(1.0 + 2.0*nu*sinh(0.5*u).^2).*chem_hold - chem_all);
136 end
137
138 % ------------------------------------------------------------
```

```
% Functions to integrate for Dp1_Dzeta
% ------------------------------------------------------------

function result = fun_nosteric_hold2(u,max,chem_all2)
chem_hold = 0;
for i = 1:size(u)
    [~,chem_hold(i)] = chemiphoresis(u(i),max,nu);
end
result = 1/2./sinh(0.5*u) .* (cosh(u).*chem_hold - chem_all2);
end

function result = fun_small_hold2(u,max,chem_all2)
chem_hold = 0;
for i = 1:size(u)
    [~,chem_hold(i)] = chemiphoresis(u(i),max,nu);
end
result = 1./abs(u).*(1./(1.0 + 1.0/2.0*nu*u.^2).*chem_hold - chem_all2);
end

function result = fun_hold2(u,max,chem_all2)
chem_hold = 0;
for i = 1:size(u)
    [~,chem_hold(i)] = chemiphoresis(u(i),max,nu);
end
result = 1./(sqrt((2.0/nu).*log(1.0 + 2.0*nu*sinh(0.5*u).^2))) .* ...
    (cosh(u)./(1.0 + 2.0*nu*sinh(0.5*u).^2).*chem_hold - chem_all2);
end

end
```

A.3.5 Calculate advection quantity p_2: peclet2.m

```
% peclet2 computes the quantity p_2 required to account for advection in
% the excess surface flux around the particle.
%
```

```
4  % function [p2, Dp2_Dzeta] = peclet2(zeta, c_s, nu, res_tol)
5  %
6  % Inputs:
7  % zeta (required):      Zeta potential at the surface of the particle
8  % c_s (required):       Bulk concentration
9  % nu (required):        Steric parameter
10 % res_tol (optional):   Tolerance for residual in computation of w
11 %
12 % Output:
13 % p2:           Desired quantity
14 % Dp2_Dzeta:    Derivative of p2 with respect to the zeta potential
15 % ------------------------------------------------------------------------------
16
17 function [p2, Dp2_Dzeta] = peclet2(zeta, c_s, nu, res_tol)
18
19 % -------------------------------------------------------------
20 % STEP 1: check argument list and set default values
21 % -------------------------------------------------------------
22
23 max_args = 4;
24 if (nargin < 3)
25     error('MATLAB:missingArgs','peclet2:missing arguments');
26 end
27 if (nargin < 4)
28     res_tol = 0.01;
29 end
30
31 % -------------------------------------------------------------
32 % STEP 2: Initialization
33 % -------------------------------------------------------------
34 num_gridpts = length(zeta);
35
36 % -------------------------------------------------------------
37 % STEP 3: Sort the values of the zeta potential by ascending order
38 % -------------------------------------------------------------
39
```

```
40 [zeta_s, idx] = sort(abs(zeta));
41
42 % ------------------------------------------------------------
43 % STEP 4: Perform the integration
44 % ------------------------------------------------------------
45
46 p2 = spalloc(num_gridpts, 1, 0);
47 Dp2_Dzeta = spalloc(num_gridpts, 1, 0);
48
49 x_min = 0.0;
50 res_w = 0.0;
51
52 for n = 1:num_gridpts
53
54     % Initialization
55     x_max = zeta_s(n);
56
57     fun_small = @(u) fun_small_hold(u,x_max);
58     fun = @(u) fun_hold(u,x_max);
59     fun_nosteric = @(u) fun_nosteric_hold(u,x_max);
60     fun_small2 = @(u) fun_small_hold2(u,x_max);
61     fun2 = @(u) fun_hold2(u,x_max);
62     fun_nosteric2 = @(u) fun_nosteric_hold2(u,x_max);
63
64     % Evaluate the integral
65     if (nu == 0)
66         res_w = res_w + quad(fun_nosteric, x_min, x_max, res_tol);
67     else
68         if (x_min < 1E-5 && x_max < 1E-5)
69             res_w = res_w + quad(fun_small, x_min, x_max, res_tol);
70         elseif (x_min < 1E-5 && x_max > 1E-5)
71             res_w = res_w + quad(fun_small, x_min, 1E-5, res_tol) + ...
72                 quad(fun, 1E-5, x_max, res_tol);
73         else
74             res_w = res_w + quad(fun, x_min, x_max, res_tol);
75         end
```

```
76      end
77
78      % Update
79      p2(n) = res_w;
80      if (nu == 0)
81          Dp2_Dzeta(n) = fun_nosteric(x_max) + ...
82              quad(fun_nosteric2, x_min, x_max, res_tol);
83      else
84          if (x_min < 1E-5 && x_max < 1E-5)
85              Dp2_Dzeta(n) = fun(x_max) + ...
86                  quad(fun_small2, x_min, x_max, res_tol);
87          elseif (x_min < 1E-5 && x_max > 1E-5)
88              Dp2_Dzeta(n) = fun(x_max) + ...
89                  quad(fun_small2, x_min, 1E-5, res_tol) + ...
90                  quad(fun2, 1E-5, x_max, res_tol);
91          else
92              Dp2_Dzeta(n) = fun(x_max) + quad(fun2, x_min, x_max, res_tol);
93          end
94      end
95      x_min = x_max;
96
97  end
98
99  % ------------------------------------------------------------
100 % STEP 5: Re-sort the arrays values
101 % ------------------------------------------------------------
102
103 p2(idx) = p2;
104 Dp2_Dzeta(idx) = Dp2_Dzeta;
105 Dp2_Dzeta = Dp2_Dzeta.*sign(zeta);
106
107 % ------------------------------------------------------------
108 % Functions to integrate for p2
109 % ------------------------------------------------------------
110
111 function result = fun_nosteric_hold(u,max)
```

```
112 result = -1/2./sinh(0.5*u) .* (cosh(u).*(max-u) - max);
113 end
114
115 function result = fun_small_hold(u,max)
116 result = -1./abs(u).*(1./(1.0 + 1.0/2.0*nu*u.^2).*(max-u) - max);
117 end
118
119 function result = fun_hold(u,max)
120 result = -1./(sqrt((2.0/nu).*log(1.0 + 2.0*nu*sinh(0.5*u).^2))) .* ...
121     (cosh(u)./(1.0 + 2.0*nu*sinh(0.5*u).^2).*(max-u) - max);
122 end
123
124 % ------------------------------------------------------------
125 % Functions to integrate for Dp2_Dzeta
126 % ------------------------------------------------------------
127
128 function result = fun_nosteric_hold2(u,max)
129 result = -1/2./sinh(0.5*u) .* (cosh(u) - 1);
130 end
131
132 function result = fun_small_hold2(u,max)
133 result = -1./abs(u).*(1./(1.0 + 1.0/2.0*nu*u.^2) - 1);
134 end
135
136 function result = fun_hold2(u,max)
137 result = -1./(sqrt((2.0/nu).*log(1.0 + 2.0*nu*sinh(0.5*u).^2))) .* ...
138     (cosh(u)./(1.0 + 2.0*nu*sinh(0.5*u).^2) - 1);
139 end
140
141 end
```

A.3.6 Calculate advection quantity p_3: peclet3.m

```
1 % peclet3 computes the quantity p_3 required to account for advection in
2 % the excess surface flux around the particle.
```

```
3  %
4  % function [p3, Dp3_Dzeta] = peclet3(zeta, c_s, nu, res_tol)
5  %
6  % Inputs:
7  % zeta (required):      Zeta potential at the surface of the particle
8  % c_s (required):       Bulk concentration
9  % nu (required):        Steric parameter
10 % res_tol (optional):   Tolerance for residual in computation of w
11 %
12 % Output:
13 % p3:           Desired quantity
14 % Dp3_Dzeta:    Derivative of p3 with respect to the zeta potential
15 % ------------------------------------------------------------------------
16
17 function [p3, Dp3_Dzeta] = peclet3(zeta, c_s, nu, res_tol)
18
19 % ---------------------------------------------------------
20 % STEP 1: check argument list and set default values
21 % ---------------------------------------------------------
22
23 max_args = 4;
24 if (nargin < 3)
25     error('MATLAB:missingArgs','peclet3:missing arguments');
26 end
27 if (nargin < 4)
28     res_tol = 0.01;
29 end
30
31 % ---------------------------------------------------------
32 % STEP 2: Initialization
33 % ---------------------------------------------------------
34 num_gridpts = length(zeta);
35
36 % ---------------------------------------------------------
37 % STEP 3: Sort the values of the zeta potential by ascending order
38 % ---------------------------------------------------------
```

```
39
40 [zeta_s, idx] = sort(abs(zeta));
41
42 % ------------------------------------------------------------
43 % STEP 4: Perform the integration
44 % ------------------------------------------------------------
45
46 p3 = spalloc(num_gridpts, 1, 0);
47 Dp3_Dzeta = spalloc(num_gridpts, 1, 0);
48
49 x_min = 0.0;
50 res_w = 0.0;
51
52 for n = 1:num_gridpts
53
54     % Initialization
55     x_max = zeta_s(n);
56
57     fun_small = @(u) fun_small_hold(u,x_max);
58     fun = @(u) fun_hold(u,x_max);
59     fun_nosteric = @(u) fun_nosteric_hold(u,x_max);
60     fun_small2 = @(u) fun_small_hold2(u,x_max);
61     fun2 = @(u) fun_hold2(u,x_max);
62     fun_nosteric2 = @(u) fun_nosteric_hold2(u,x_max);
63
64     % Evaluate the integral
65     if (nu == 0)
66         res_w = res_w + quad(fun_nosteric, x_min, x_max, res_tol);
67     else
68         if (x_min < 1E-5 && x_max < 1E-5)
69             res_w = res_w + quad(fun_small, x_min, x_max, res_tol);
70         elseif (x_min < 1E-5 && x_max > 1E-5)
71             res_w = res_w + quad(fun_small, x_min, 1E-5, res_tol) + ...
72                 quad(fun, 1E-5, x_max, res_tol);
73         else
74             res_w = res_w + quad(fun, x_min, x_max, res_tol);
```

```
75          end
76      end
77
78      % Update
79      p3(n) = res_w;
80      if (nu == 0)
81          Dp3_Dzeta(n) = fun_nosteric(x_max) + ...
82              quad(fun_nosteric2, x_min, x_max, res_tol);
83      else
84          if (x_min < 1E-5 && x_max < 1E-5)
85              Dp3_Dzeta(n) = fun(x_max) + ...
86                  quad(fun_small2, x_min, x_max, res_tol);
87          elseif (x_min < 1E-5 && x_max > 1E-5)
88              Dp3_Dzeta(n) = fun(x_max) + ...
89                  quad(fun_small2, x_min, 1E-5, res_tol) + ...
90                  quad(fun2, 1E-5, x_max, res_tol);
91          else
92              Dp3_Dzeta(n) = fun(x_max) + quad(fun2, x_min, x_max, res_tol);
93          end
94      end
95      x_min = x_max;
96
97  end
98
99  % ------------------------------------------------------------
100 % STEP 5: Re-sort the arrays values
101 % ------------------------------------------------------------
102
103 p3(idx) = p3;
104 Dp3_Dzeta(idx) = Dp3_Dzeta;
105 Dp3_Dzeta = Dp3_Dzeta.*sign(zeta);
106
107 % ------------------------------------------------------------
108 % Functions to integrate for p3
109 % ------------------------------------------------------------
110
```

```
111 function result = fun_nosteric_hold(u,max)
112 chem_hold = 0;
113 for i = 1:size(u)
114     chem_hold(i) = chemiphoresis(u(i),max,nu);
115 end
116 result = -1/2./sinh(0.5*u) .* (sinh(u).*chem_hold);
117 end
118
119 function result = fun_small_hold(u,max)
120 chem_hold = 0;
121 for i = 1:size(u)
122     chem_hold(i) = chemiphoresis(u(i),max,nu);
123 end
124 result = -1./abs(u).*(u./(1.0 + 1.0/2.0*nu*u.^2).*chem_hold);
125 end
126
127 function result = fun_hold(u,max)
128 chem_hold = 0;
129 for i = 1:size(u)
130     chem_hold(i) = chemiphoresis(u(i),max,nu);
131 end
132 result = -1./(sqrt((2.0/nu).*log(1.0 + 2.0*nu*sinh(0.5*u).^2))) .* ...
133     (sinh(u)./(1.0 + 2.0*nu*sinh(0.5*u).^2).*chem_hold);
134 end
135
136 % ------------------------------------------------------------
137 % Functions to integrate for Dp3_Dzeta
138 % ------------------------------------------------------------
139
140 function result = fun_nosteric_hold2(u,max)
141 chem_hold = 0;
142 for i = 1:size(u)
143     [~,chem_hold(i)] = chemiphoresis(u(i),max,nu);
144 end
145 result = -1/2./sinh(0.5*u) .* (sinh(u).*chem_hold);
146 end
```

```
147
148 function result = fun_small_hold2(u,max)
149 chem_hold = 0;
150 for i = 1:size(u)
151     [~,chem_hold(i)] = chemiphoresis(u(i),max,nu);
152 end
153 result = -1./abs(u).*(u./(1.0 + 1.0/2.0*nu*u.^2).*chem_hold);
154 end
155
156 function result = fun_hold2(u,max)
157 chem_hold = 0;
158 for i = 1:size(u)
159     [~,chem_hold(i)] = chemiphoresis(u(i),max,nu);
160 end
161 result = -1./(sqrt((2.0/nu).*log(1.0 + 2.0*nu*sinh(0.5*u).^2))) .* ...
162     (sinh(u)./(1.0 + 2.0*nu*sinh(0.5*u).^2).*chem_hold);
163 end
164
165 end
```

A.3.7 Calculate advection quantity p_4: peclet4.m

```
1  % peclet4 computes the quantity p_4 required to account for advection in
2  % the excess surface flux around the particle.
3  %
4  % function [p4, Dp4_Dzeta] = peclet4(zeta, c_s, nu, res_tol)
5  %
6  % Inputs:
7  % zeta (required):      Zeta potential at the surface of the particle
8  % c_s (required):       Bulk concentration
9  % nu (required):        Steric parameter
10 % res_tol (optional):   Tolerance for residual in computation of w
11 %
12 % Output:
13 % p4:           Desired quantity
```

```
14 % Dp4_Dzeta:    Derivative of p4 with respect to the zeta potential
15 % ------------------------------------------------------------------------------
16
17 function [p4, Dp4_Dzeta] = peclet4(zeta, c_s, nu, res_tol)
18
19 % ------------------------------------------------------------
20 % STEP 1: check argument list and set default values
21 % ------------------------------------------------------------
22
23 max_args = 4;
24 if (nargin < 3)
25     error('MATLAB:missingArgs','peclet4:missing arguments');
26 end
27 if (nargin < 4)
28     res_tol = 0.01;
29 end
30
31 % ------------------------------------------------------------
32 % STEP 2: Initialization
33 % ------------------------------------------------------------
34 num_gridpts = length(zeta);
35
36 % ------------------------------------------------------------
37 % STEP 3: Sort the values of the zeta potential by ascending order
38 % ------------------------------------------------------------
39
40 [zeta_s, idx] = sort(abs(zeta));
41
42 % ------------------------------------------------------------
43 % STEP 4: Perform the integration
44 % ------------------------------------------------------------
45
46 p4 = spalloc(num_gridpts, 1, 0);
47 Dp4_Dzeta = spalloc(num_gridpts, 1, 0);
48
49 x_min = 0.0;
```

```
50 res_w = 0.0;
51
52 for n = 1:num_gridpts
53
54     % Initialization
55     x_max = zeta_s(n);
56
57     fun_small = @(u) fun_small_hold(u,x_max);
58     fun = @(u) fun_hold(u,x_max);
59     fun_nosteric = @(u) fun_nosteric_hold(u,x_max);
60     fun_small2 = @(u) fun_small_hold2(u,x_max);
61     fun2 = @(u) fun_hold2(u,x_max);
62     fun_nosteric2 = @(u) fun_nosteric_hold2(u,x_max);
63
64     % Evaluate the integral
65     if (nu == 0)
66         res_w = res_w + quad(fun_nosteric, x_min, x_max, res_tol);
67     else
68         if (x_min < 1E-5 && x_max < 1E-5)
69             res_w = res_w + quad(fun_small, x_min, x_max, res_tol);
70         elseif (x_min < 1E-5 && x_max > 1E-5)
71             res_w = res_w + quad(fun_small, x_min, 1E-5, res_tol) + ...
72                 quad(fun, 1E-5, x_max, res_tol);
73         else
74             res_w = res_w + quad(fun, x_min, x_max, res_tol);
75         end
76     end
77
78     % Update
79     p4(n) = res_w;
80     if (nu == 0)
81         Dp4_Dzeta(n) = fun_nosteric(x_max) + ...
82             quad(fun_nosteric2, x_min, x_max, res_tol);
83     else
84         if (x_min < 1E-5 && x_max < 1E-5)
85             Dp4_Dzeta(n) = fun(x_max) + ...
```

```
86                  quad(fun_small2, x_min, x_max, res_tol);
87          elseif (x_min < 1E-5 && x_max > 1E-5)
88              Dp4_Dzeta(n) = fun(x_max) + ...
89                  quad(fun_small2, x_min, 1E-5, res_tol) + ...
90                  quad(fun2, 1E-5, x_max, res_tol);
91          else
92              Dp4_Dzeta(n) = fun(x_max) + quad(fun2, x_min, x_max, res_tol);
93          end
94      end
95      x_min = x_max;
96
97  end
98
99  % ------------------------------------------------------------
100 % STEP 5: Re-sort the arrays values
101 % ------------------------------------------------------------
102
103 p4(idx) = p4;
104 Dp4_Dzeta(idx) = Dp4_Dzeta;
105 Dp4_Dzeta = Dp4_Dzeta.*sign(zeta);
106
107 % ------------------------------------------------------------
108 % Functions to integrate for p4
109 % ------------------------------------------------------------
110
111 function result = fun_nosteric_hold(u,max)
112 result = 1/2./sinh(0.5*u) .* (sinh(u).*(max-u));
113 end
114
115 function result = fun_small_hold(u,max)
116 result = 1./abs(u).*(u./(1.0 + 1.0/2.0*nu*u.^2).*(max-u));
117 end
118
119 function result = fun_hold(u,max)
120 result = 1./(sqrt((2.0/nu).*log(1.0 + 2.0*nu*sinh(0.5*u).^2))) .* ...
121     (sinh(u)./(1.0 + 2.0*nu*sinh(0.5*u).^2).*(max-u));
```

```
122 end
123
124 % ------------------------------------------------------------
125 % Functions to integrate for Dp4_Dzeta
126 % ------------------------------------------------------------
127
128 function result = fun_nosteric_hold2(u,max)
129 result = 1/2./sinh(0.5*u) .* (sinh(u));
130 end
131
132 function result = fun_small_hold2(u,max)
133 result = 1./abs(u).*(u./(1.0 + 1.0/2.0*nu*u.^2));
134 end
135
136 function result = fun_hold2(u,max)
137 result = 1./(sqrt((2.0/nu).*log(1.0 + 2.0*nu*sinh(0.5*u).^2))) .* ...
138     (sinh(u)./(1.0 + 2.0*nu*sinh(0.5*u).^2));
139 end
140
141 end
```

A.4 Main program (Steric model)

When Pe = 0, all references to the variables p_i could be removed from the code to reduce computation time.

A.4.1 solveHighFieldSteadyResponse3D.m

```
1 % Adapted from Kevin Chu's PhD thesis / 2005 (Dept of Mathematics, MIT)
2 %
3 % solveHighFieldSteadyResponse3D solves for the steady response of a
4 % spherical particle in the strongly nonlinear limit. The unknowns are
5 % taken to be the concentration, c, the potential relative to the applied
6 % potential, psi, and the stream Stokes function phi.
```

```
7  %
8  % Usage: function [c, psi, phi, u_r, u_theta, V, zeta, u_s, u_sc, q, w, ...
9  %           c_s, U, r, theta] = solveHighFieldSteadyResponse3D( ...
10 %           Q, E, epsilon, c_infinity, nu, peclet, ...
11 %           N_r, L_r, N_theta, ...
12 %           c_init, psi_init, phi_init, U_init, ...
13 %           E_start, dE, ...
14 %           res_tol, delta_tol, max_iters, ...
15 %           show_stats, ...
16 %           zeta_res_tol, zeta_delta_tol, zeta_max_iters)
17 %
18 % Inputs:
19 % Q (required):             charge of the particle
20 % E (required):             applied electric field
21 % epsilon (required):       ratio of Debye length to sphere radius
22 % c_infinity (required):    boundary condition for concentration at r = infty
23 % nu (required):            packing parameter
24 % peclet (required):        Peclet number
25 % N_r (required):           order of approximation in radial direction
26 % L_r (required):           scale parameter in radial direction
27 % N_theta (required):       order of approximation in polar angle direction
28 % c_init (optional):        initial iterate for c
29 % psi_init (optional):      initial iterate for psi
30 % phi_init (optional):      initial iterate for phi
31 % U_init (optional):        initial iterate for U
32 % E_start (optional):       starting value for E continuation
33 % dE (optional):            size of steps for E continuation
34 % res_tol (optional):       tolerance for residual in Newton iteration
35 % delta_tol (optional):     tolerance for change in Newton iteration
36 % max_iters (optional):     maximum number of iterations for Newton iteration
37 % show_stats (optional):    show statistics from psi/c claculation
38 % zeta_res_tol (optional):  tolerance for residual in zeta calculation
39 % zeta_delta_tol (optional):tolerance for change in zeta calculation
40 % zeta_max_iters (optional):maximum number of iterations for zeta calculation
41 %
42 % Outputs:
```

```
% c:             concentration profile
% psi:           electric potential relative to background applied potential
% phi:           Stokes stream function
% u_r:           Radial velocity
% u_theta:       Angular velocity
% V:             Potential of the spherical particle
% zeta:          Zeta potential
% u_s:           Slip velocity
% u_sc:          Diffusio-osmotic velocity
% q:             Excess charge
% w:             Excess salt
% U:             Electrophoretic velocity
% r:             Radial grid points
% theta:         Polar grid points
%
% NOTES:
% (1) The direction of the normal to the surface of the sphere is OUTWARD
% from the physical domain. That is, it points INTO the sphere.
% ------------------------------------------------------------------------

function [c, psi, phi, u_r, u_theta, V, zeta, u_s, u_sc, q, w, ...
    c_s, U, r, theta] = solveHighFieldSteadyResponse3D( ...
    Q, E, epsilon, c_infinity, nu, peclet, ...
    N_r, L_r, N_theta, ...
    c_init, psi_init, phi_init, U_init, ...
    E_start, dE, ...
    res_tol, delta_tol, max_iters, ...
    show_stats, ...
    zeta_res_tol, zeta_delta_tol, zeta_max_iters)

% ------------------------------------------------------------------------
% STEP 1: check argument list and set default values
% ------------------------------------------------------------------------

max_args = 22;
if (nargin < 9)
```

```
79      error('MATLAB:missingArgs', 's3D:missing arguments');
80  end
81  if (nargin < max_args)
82      zeta_max_iters = 10;
83  end
84  if (nargin < max_args - 1)
85      zeta_delta_tol = 1E-13;
86  end
87  if (nargin < max_args - 2)
88      zeta_res_tol = 1E-9;
89  end
90  if (nargin < max_args - 3)
91      show_stats = 0;
92  end
93  if (nargin < max_args - 4)
94      max_iters = 12;
95  end
96  if (nargin < max_args - 5)
97      delta_tol = 1E-4;
98  end
99  if (nargin < max_args - 6)
100     res_tol = 5E-4;
101 end
102 if (nargin < max_args - 8)
103     E_continuation = 0;
104     dE = 1;
105 else
106     E_continuation = 1;
107     if (nargin < max_args - 7)
108         warning('MATLAB:badopt', ...
109             's3D: no step size for E continuation ... using dE = 0.1');
110         dE = 0.1;
111     else
112         if (dE == 0)
113             E_continuation = 0;
114             dE = 1;
```

```
115         end
116     end
117 end
118 if (nargin < max_args - 9)
119     generate_initial_guess = 1;
120 else
121         if (length(c_init) == (N_r+1)*N_theta && ...
122             length(psi_init) == (N_r+1)*N_theta && ...
123             length(phi_init) == (N_r+1)*N_theta && ...
124             length(U_init) == 1)
125             c = c_init;
126             psi = psi_init;
127             phi = phi_init;
128             U = U_init;
129             generate_initial_guess = 2;
130         elseif (length(c_init) ~= N_r*N_theta || ...
131                 length(psi_init) ~= N_r*N_theta || ...
132                 length(phi_init) ~= N_r*N_theta || ...
133                 length(U_init) ~= 1)
134             warning('MATLAB:badopt', ...
135                 's3D: initial guess inconsistent with grid size');
136             generate_initial_guess = 1;
137         else
138             c = c_init;
139             psi = psi_init;
140             phi = phi_init;
141             U = U_init;
142             generate_initial_guess = 0;
143         end
144 end
145
146 % setup E continuation parameters
147 if (E_continuation ~= 0)
148     E_end = E;
149     if (E < E_start)
150         dE = -dE;
```

```
151     end
152 else
153     E_start = E;
154     E_end = E;
155 end
156
157 % set up E_values to use in continuation making sure it ends with E_end
158 E_values = E_start:dE:E_end;
159 if (E_values(end) ~= E_end)
160     E_values = [E_values E_end];
161 end
162
163 % ------------------------------------------------------------------------
164 % STEP 2: construct computational grid and differentiation operators
165 % ------------------------------------------------------------------------
166
167 % -----
168 % 1: Operators definition
169 % -----
170
171 % Construct the differentiation matrices for the polar / radial coordinates
172 [D_r, r] = DM_TL(N_r, L_r);
173 r = r + 1;
174 Dr = -diff(r);
175 Dr = Dr(2:end);
176 [D_theta, theta, Dtheta] = DM_cosine_interior(N_theta);
177 DrInt=kron(ones(N_theta,1),Dr);
178 DthetaInt=kron(Dtheta',ones(N_r-1,1));
179 OnesInt=ones((N_r-1)*N_theta,1);
180
181 % Define some common expressions for the number of grid points
182 num_gridpts_r = length(r);
183 num_gridpts_theta = length(theta);
184 num_gridpts = (num_gridpts_r - 1)*N_theta;
185 num_gridpts_interior = num_gridpts - N_theta;
186 num_gridpts_stokes = num_gridpts_interior - N_theta;
```

```
187
188 % Define some cache common expressions
189 cos_theta = cos(theta);
190 sin_theta = sin(theta);
191
192 cos_theta_full = kron(cos_theta, ones(num_gridpts_r - 1, 1));
193 sin_theta_full = kron(sin_theta, ones(num_gridpts_r - 1, 1));
194
195 cos_theta_interior = kron(cos_theta, ones(num_gridpts_r - 2, 1));
196 sin_theta_interior = kron(sin_theta, ones(num_gridpts_r - 2, 1));
197
198 r_interior = r(3:end);
199 one_r_interior = kron(ones(N_theta,1),1./r_interior);
200
201 % construct divergence, gradient, Laplacian and Stokes operators
202 D = div(D_r, D_theta, r, theta);
203 G = grad(D_r, D_theta, r, theta);
204 L = laplacian(D_r, D_theta, r, theta);
205 [S, P] = stokes(G, D_r, D_theta, r, theta);
206
207 % construct surface divergence operators
208 D_s = div_s(D_theta, theta, r(end));
209 G_s = D_theta./r(end);
210
211 % construct normal derivative operator
212 G_n = -kron(speye(N_theta),D_r(end,:));
213
214 % -----
215 % 2: Rows and columns extraction tools
216 % -----
217
218 % construct matrix to extract the rows corresponding to r = infty
219 r_inf_extractor = spalloc(1, num_gridpts_r, 1);
220 r_inf_extractor(1,1) = 1;
221 inf_extractor = kron(speye(N_theta), r_inf_extractor);
222
```

```
223 % construct matrix to extract the rows corresponding to finite grid points
224 % (everything except for r = infty)
225 r_finite_pt_extractor = spdiags(ones(num_gridpts_r - 1, 1), 1, ...
226     num_gridpts_r - 1, num_gridpts_r);
227 finite_pt_extractor = kron(speye(N_theta), r_finite_pt_extractor);
228
229 % construct matrix to extract the rows corresponding to interior grid points
230 % (everything except for r = 1 and r = infty)
231 r_interior_extractor = spdiags(ones(num_gridpts_r - 2, 1), 1, ...
232     num_gridpts_r - 2, num_gridpts_r);
233 interior_extractor = kron(speye(N_theta), r_interior_extractor);
234
235 % construct matrix to extract the rows corresponding to r = 1 (surface)
236 % from a vector that already has r = infinity removed
237 r_surf_extractor = spalloc(1, num_gridpts_r - 1, 1);
238 r_surf_extractor(1, num_gridpts_r - 1) = 1;
239 surf_extractor = kron(speye(N_theta), r_surf_extractor);
240
241 % Construct matrix to extract the rows corresponding to interior grid points
242 % (everything except for r = 1, 2 and r = infty), used for the Stokes operator
243 r_stokes_extractor = spdiags(ones(num_gridpts_r - 3, 1), 1, ...
244     num_gridpts_r - 3, num_gridpts_r);
245 stokes_extractor = kron(speye(N_theta), r_stokes_extractor);
246
247 % -----
248 % 2: Operators splitting
249 % -----
250
251 % split the G operator
252 G_f = {finite_pt_extractor*G{1}*finite_pt_extractor', ...
253     finite_pt_extractor*G{2}*finite_pt_extractor'};
254 G_i = {interior_extractor*G{1}*finite_pt_extractor', ...
255     interior_extractor*G{2}*finite_pt_extractor'};
256 G_stokes = {stokes_extractor*G{1}*finite_pt_extractor', ...
257     stokes_extractor*G{2}*finite_pt_extractor'};
258 G_inf = {interior_extractor*G{1}*inf_extractor', ...
```

```
259     interior_extractor*G{2}*inf_extractor'};
260 G_inf_f = {finite_pt_extractor*G{1}*inf_extractor', ...
261     finite_pt_extractor*G{2}*inf_extractor'};
262
263 % split the D operator
264 D_f = {finite_pt_extractor*D{1}*finite_pt_extractor', ...
265     finite_pt_extractor*D{2}*finite_pt_extractor'};
266 D_i = { interior_extractor*D{1}*finite_pt_extractor', ...
267     interior_extractor*D{2}*finite_pt_extractor'};
268 D_inf = { interior_extractor*D{1}*inf_extractor', ...
269     interior_extractor*D{2}*inf_extractor'};
270
271 % split the G_n operator
272 G_n_f = G_n * finite_pt_extractor';
273 G_n_inf = G_n * inf_extractor';
274
275 % split the Laplacian operator
276 L_f = finite_pt_extractor*L*finite_pt_extractor';
277 L_i = interior_extractor*L*finite_pt_extractor';
278 L_inf = interior_extractor*L*inf_extractor';
279
280 % split the Stokes operator
281 S_stokes = stokes_extractor*S*finite_pt_extractor';
282 S_inf = stokes_extractor*S*inf_extractor';
283
284 % split the Stokes function operator
285 P_i = {interior_extractor*P{1}*finite_pt_extractor', ...
286     interior_extractor*P{2}*finite_pt_extractor'};
287 P_f = {finite_pt_extractor*P{1}*finite_pt_extractor', ...
288     finite_pt_extractor*P{2}*finite_pt_extractor'};
289 P_surf = {surf_extractor*P_f{1}, surf_extractor*P_f{2}};
290 P_inf = {interior_extractor*P{1}*inf_extractor', ...
291     interior_extractor*P{2}*inf_extractor'};
292 P_inf_f = {finite_pt_extractor*P{1}*inf_extractor', ...
293     finite_pt_extractor*P{2}*inf_extractor'};
294
```

```
295 % clear auxilliary operators
296 clear D G L G_n S
297
298 % ------------------------------------------------------------------------
299 % STEP 3: generate initial guess if required c = 1, psi = 0, phi = 0, U = 0
300 % ------------------------------------------------------------------------
301 if (generate_initial_guess == 2)
302     c = finite_pt_extractor*c;
303     psi = finite_pt_extractor*psi;
304     phi = finite_pt_extractor*phi;
305     generate_initial_guess = 0;
306 end
307
308 if (generate_initial_guess)
309     c = c_infinity * ones(num_gridpts, 1);
310     psi = zeros(num_gridpts, 1);
311     phi = zeros(num_gridpts, 1);
312     U = 0;
313 end
314
315 % ------------------------------------------------------------------------
316 % STEP 4: continuation loop for Newton iteration
317 % ------------------------------------------------------------------------
318
319 count_total = 0;
320
321 % begin loop over E_values
322 for E = E_values
323
324     if (show_stats > 1)
325         mesg = sprintf('E = %f', E);
326         disp(mesg);
327     end
328
329     % ------------------------------------------------------------------------
330     % step 1: compute constant terms in F = (F1, F2, F3, H1, H2, H3)
```

```
% ------------------------------------------------------------------------

% Bulk equations
F1_const_term = c_infinity * (L_inf * ones(num_gridpts_theta, 1));
F2_const_term = E*c_infinity * (-D_inf{1}*cos_theta + D_inf{2}*sin_theta);

% Boundary conditions
H2_const_term = - c_infinity * (G_n_inf*ones(num_gridpts_theta, 1));

% ------------------------------------------------------------------------
% Step 2: compute constant parts of Jacobian
% ------------------------------------------------------------------------

% Bulk equations
DF1_Dc_const = L_i;

DF1_Dphi_const =  - peclet*( ...
spdiags(c_infinity * (G_inf{1} * ones(num_gridpts_theta, 1)), 0, ...
num_gridpts_interior, num_gridpts_interior ) * P_i{1} + ...
spdiags(c_infinity * (G_inf{2} * ones(num_gridpts_theta, 1)), 0, ...
num_gridpts_interior, num_gridpts_interior ) * P_i{2} ...
);

DF1_DU_const =  - peclet*( ...
spdiags(c_infinity * (G_inf{1} * ones(num_gridpts_theta, 1)), 0, ...
num_gridpts_interior, num_gridpts_interior ) * ...
(P_inf{1} * ones(num_gridpts_theta, 1)) + ...
spdiags(c_infinity * (G_inf{2} * ones(num_gridpts_theta, 1)), 0, ...
num_gridpts_interior, num_gridpts_interior ) * ...
(P_inf{2} * ones(num_gridpts_theta, 1)) ...
);

DF2_Dc_const = - E * D_i{1} * spdiags(cos_theta_full, 0, ...
    num_gridpts, num_gridpts) + ...
    E * D_i{2} * spdiags(sin_theta_full, 0, num_gridpts, num_gridpts);

```

```
367     DF3_Dphi_const = S_stokes;
368     DF3_DU_const = (S_inf * ones(num_gridpts_theta, 1));
369
370     % Momentum conservation
371     DM1_DU_const = - 2.0;
372
373     % Boundary conditions
374     DH1_Dc_const = spalloc(num_gridpts_theta, num_gridpts, num_gridpts_theta);
375     DH1_Dc_const(:, num_gridpts_r - 1:num_gridpts_r - 1:end) = ...
376         - E * spdiags(cos_theta, 0, num_gridpts_theta, num_gridpts_theta);
377
378     DH2_Dc_const = - G_n_f;
379
380     DH3_Dphi_const = eye(N_theta) * surf_extractor;
381     DH4_Dphi_const = P_surf{2};
382
383     % ------------------------------------------------------------------------
384     % Step 3: initialize loop variables using current solution for c and psi
385     % ------------------------------------------------------------------------
386
387     % Extract surface concentration and potential
388     c_s = c(num_gridpts_r - 1:num_gridpts_r - 1:end);
389     psi_s = psi(num_gridpts_r - 1:num_gridpts_r - 1:end) - E * cos_theta;
390
391     % Compute zeta potential
392     [V, zeta, q, w, Dzeta_Dpsi, Dzeta_Dc_s, Dq_Dpsi, Dq_Dc_s, Dw_Dpsi, ...
393         Dw_Dc_s, p1, p2, p3, p4, Dp1_Dpsi, Dp1_Dc_s, Dp2_Dpsi, Dp2_Dc_s, ...
394         Dp3_Dpsi, Dp3_Dc_s, Dp4_Dpsi, Dp4_Dc_s]...
395         = computeZetaPotential(psi_s, c_s, nu, Q);
396
397     % Chemiphoresis
398     [u_sc, Du_sc_Dzeta] = chemiphoresis(0, zeta, nu);
399
400     % Compute initial residual for the bulk equations
401     F1 = F1_const_term + DF1_Dc_const*c - peclet * ( ...
402     (G_i{1}*c + c_infinity * (G_inf{1} * ones(num_gridpts_theta, 1)) ) .* ...
```

```
403     (P_i{1} * phi + U * (P_inf{1} * ones(num_gridpts_theta, 1)) ) + ...
404         (G_i{2}*c + c_infinity * (G_inf{2} * ones(num_gridpts_theta, 1)) ) .* ...
405         (P_i{2} * phi + U * (P_inf{2} * ones(num_gridpts_theta, 1)) ) ...
406     );
407
408     F2 = F2_const_term + DF2_Dc_const*c + D_i{1} * (c .* (G_f{1} * psi)) + ...
409         D_i{2} * (c .* (G_f{2} * psi));
410     F3 = DF3_Dphi_const*phi + DF3_DU_const*U + (G_stokes{1} * L_f * psi) .* ...
411         (G_stokes{2} * psi) - (G_stokes{2} * L_f * psi) .* (G_stokes{1} * psi);
412
413     % Momentum conservation equation
414     Maxwell = (-1/3 * (1/2 * (G_i{1}*psi).^2 .* cos_theta_interior .* ...
415         (-9/4.*one_r_interior + 9/4.*one_r_interior.^3) ...
416         - 1/2 * (G_i{2}*psi).^2 .* cos_theta_interior .* ...
417         (-9/4.*one_r_interior + 9/4.*one_r_interior.^3) ...
418         + (G_i{1}*psi) .* (G_i{2}*psi) .* sin_theta_interior .* ...
419         (3/2.*one_r_interior.^3)) .* DrInt .* DthetaInt)' * OnesInt;
420     M1 = DM1_DU_const*U + Dtheta * ( zeta .* (G_s * psi_s) ...
421         - u_sc .* (G_s * c_s) ./ c_s ) + Maxwell;
422
423     % Compute initial residual for the boundary conditions
424     H1 = DH1_Dc_const * c + epsilon * D_s * ...
425         ((q + p3*peclet) ./ c_s .* (G_s * c_s) + ...
426         (w + p4*peclet) .* (G_s * psi_s)) - c_s .* (G_n_f * psi);
427     H2 = H2_const_term + DH2_Dc_const * c + epsilon * D_s * ...
428         ((w + p1*peclet) ./ c_s .* (G_s * c_s) + ...
429         (q + p2*peclet) .* (G_s * psi_s));
430     H3 = DH3_Dphi_const * phi;
431     H4 = DH4_Dphi_const * phi -  zeta .* (G_s * psi_s) + ...
432         u_sc .* (G_s * c_s) ./ c_s;
433
434     % Initialization
435     F = [F1; F2; H1; H2; F3; M1; H3; H4];
436     res = norm (F, inf);
437
438
```

```
439     % ------------------------------------------------------------------------
440     % step 4: Newton iteration loop
441     % ------------------------------------------------------------------------
442
443     % initialize delta_soln and count
444     norm_delta_soln = 1;
445     count = 0;
446
447     % begin Newton iteration loop
448     while (res > res_tol && norm_delta_soln > delta_tol && count < max_iters)
449
450     % -----
451     % 1: Jacobian of the bulk equations
452     % -----
453
454     % First bulk equation
455     DF1_Dc_var = -  peclet*( ...
456         spdiags(P_i{1}*phi + U*(P_inf{1} * ones(num_gridpts_theta, 1)), ...
457         0, num_gridpts_interior, num_gridpts_interior) * G_i{1} +  ...
458         spdiags(P_i{2}*phi + U * (P_inf{2} * ones(num_gridpts_theta, 1)), ...
459         0, num_gridpts_interior, num_gridpts_interior) * G_i{2}...
460             );
461
462     DF1_Dphi_var = -peclet*( ...
463         spdiags(G_i{1}*c, 0, ...
464         num_gridpts_interior, num_gridpts_interior)*P_i{1} +  ...
465         spdiags(G_i{2}*c, 0, ...
466         num_gridpts_interior, num_gridpts_interior)*P_i{2} ...
467             );
468
469     DF1_DU_var = -  peclet*( ...
470         spdiags(G_i{1}*c, 0, ...
471         num_gridpts_interior, num_gridpts_interior) * ...
472         (P_inf{1}*ones(num_gridpts_theta, 1)) +  ...
473         spdiags(G_i{2}*c, 0, ...
474         num_gridpts_interior, num_gridpts_interior) * ...
```

```
475         (P_inf{2}*ones(num_gridpts_theta, 1)) ...
476             );
477
478
479     % Second bulk equation
480     DF2_Dc_var = D_i{1} * spdiags(G_f{1} * psi, 0, ...
481         num_gridpts, num_gridpts) + ...
482         D_i{2} * spdiags(G_f{2} * psi, 0, num_gridpts, num_gridpts);
483     DF2_Dpsi_var = D_i{1} * spdiags(c, 0, ...
484         num_gridpts, num_gridpts) * G_f{1} +  ...
485         D_i{2} * spdiags(c, 0, num_gridpts, num_gridpts) * G_f{2};
486
487     % Third bulk equation
488     DF3_Dpsi_var =  spdiags(G_stokes{1}*L_f*psi, 0, ...
489         num_gridpts_stokes, num_gridpts_stokes) * G_stokes{2} + ...
490         spdiags(G_stokes{2}*psi, 0, ...
491         num_gridpts_stokes, num_gridpts_stokes) * G_stokes{1}*L_f - ...
492         spdiags(G_stokes{2}*L_f*psi, 0,...
493         num_gridpts_stokes, num_gridpts_stokes) * G_stokes{1} - ...
494         spdiags(G_stokes{1}*psi, 0, ...
495         num_gridpts_stokes, num_gridpts_stokes) * G_stokes{2}*L_f;
496
497     % -----
498     % 2: Jacobian of the momentum conservation equation
499     % -----
500
501     % Fourth boundary condition
502     DM1_Dc_var = Dtheta * (spdiags(Dzeta_Dc_s .* (G_s * psi_s), 0, ...
503         num_gridpts_theta, num_gridpts_theta) * surf_extractor - ...
504         ( spdiags(Du_sc_Dzeta .* Dzeta_Dc_s .* (G_s * c_s) ./ c_s, 0, ...
505         num_gridpts_theta, num_gridpts_theta) ...
506         - spdiags(u_sc .* (G_s * c_s) ./ (c_s .* c_s), 0, ...
507         num_gridpts_theta, num_gridpts_theta) ...
508         + spdiags(u_sc ./ c_s , 0, num_gridpts_theta, num_gridpts_theta) ...
509         * G_s ) * surf_extractor ...
510         );
```

```
511
512     DMaxwell_Dpsi = (DrInt .* DthetaInt)' * ...
513         (-1/3 * (spdiags(((G_i{1}*psi) ...
514         .* cos_theta_interior .* ...
515         (-9/4.*one_r_interior + 9/4.*one_r_interior.^3)), ...
516         0, num_gridpts_interior, num_gridpts_interior) * G_i{1} ...
517         - (spdiags(((G_i{2}*psi) .* cos_theta_interior .* ...
518         (-9/4.*one_r_interior + 9/4.*one_r_interior.^3)), 0, ...
519         num_gridpts_interior, num_gridpts_interior) * G_i{2}) ...
520         + (spdiags(((G_i{2}*psi) .* sin_theta_interior .* ...
521         (3/2.*one_r_interior.^3)), 0, ...
522         num_gridpts_interior, num_gridpts_interior) * G_i{1}) ...
523         + (spdiags(((G_i{1}*psi) .* sin_theta_interior .* ...
524         (3/2.*one_r_interior.^3)), 0, ...
525         num_gridpts_interior, num_gridpts_interior) * G_i{2})));
526
527     DM1_Dpsi_var = Dtheta * ( (spdiags(Dzeta_Dpsi .* (G_s * psi_s), 0, ...
528         num_gridpts_theta, num_gridpts_theta)...
529         + spdiags(zeta, 0, num_gridpts_theta, num_gridpts_theta) * G_s) ...
530         * surf_extractor + ...
531         spdiags(Du_sc_Dzeta .* Dzeta_Dpsi .* (G_s * c_s) ./ c_s, 0, ...
532         num_gridpts_theta, num_gridpts_theta) * surf_extractor ...
533         ) + DMaxwell_Dpsi;
534
535     % -----
536     % 3: Jacobian of the boundary conditions
537     % -----
538
539     % First boundary condition
540     DH1_Dc_var = (epsilon * D_s * (...
541         spdiags((q + p3*peclet) ./ c_s, 0, N_theta, N_theta) * G_s ...
542         - spdiags((q + p3*peclet) .*(G_s * c_s) ./ (c_s .* c_s), ...
543         0, N_theta, N_theta) ...
544         + spdiags( (G_s * c_s) ./ c_s .* (Dq_Dc_s + Dp3_Dc_s*peclet), ...
545         0, N_theta, N_theta) ...
546         + spdiags((G_s * psi_s) .* (Dw_Dc_s + Dp4_Dc_s*peclet), ...
```

```
547         0, N_theta, N_theta) ...
548         ) ...
549         - spdiags(G_n_f * psi, 0, N_theta, N_theta) ) * surf_extractor;
550
551     DH1_Dpsi_var = epsilon * D_s * ( ...
552         spdiags( (G_s * c_s) ./ c_s .* (Dq_Dpsi + Dp3_Dpsi*peclet), ...
553         0, N_theta, N_theta ) ...
554         + spdiags((w + p4*peclet), 0, N_theta, N_theta) * G_s ...
555         + spdiags((G_s * psi_s) .* (Dw_Dpsi + Dp4_Dpsi*peclet), ...
556         0, N_theta, N_theta)...
557         ) * surf_extractor ...
558         - spdiags(c_s, 0, N_theta, N_theta) * G_n_f;
559
560     % Second boundary condition
561     DH2_Dc_var = epsilon * D_s * (...
562         spdiags((w + p1*peclet) ./ c_s, 0, N_theta, N_theta) * G_s ...
563         - spdiags((w + p1*peclet) .*(G_s * c_s) ./ (c_s .* c_s), ...
564         0, N_theta, N_theta) ...
565         + spdiags((G_s * c_s) ./ c_s .* (Dw_Dc_s + Dp1_Dc_s*peclet), ...
566         0, N_theta, N_theta) ...
567         + spdiags((G_s * psi_s) .* (Dq_Dc_s + Dp2_Dc_s*peclet), ...
568         0, N_theta, N_theta) ...
569         ) * surf_extractor;
570
571     DH2_Dpsi_var = epsilon * D_s * ( ...
572         spdiags( (G_s * c_s) ./ c_s .* (Dw_Dpsi + Dp1_Dpsi*peclet), ...
573         0, N_theta, N_theta ) ...
574         + spdiags((q + p2*peclet), 0, N_theta, N_theta) * G_s ...
575         + spdiags((G_s * psi_s) .* (Dq_Dpsi + Dp2_Dpsi*peclet), ...
576         0, N_theta, N_theta)...
577         ) * surf_extractor;
578
579     % Fourth boundary condition
580     DH4_Dc_var = spdiags(- Dzeta_Dc_s .* (G_s * psi_s), 0, ...
581         num_gridpts_theta, num_gridpts_theta) * surf_extractor + ...
582        ( spdiags(Du_sc_Dzeta .* Dzeta_Dc_s .* ...
```

```
583         (G_s * c_s) ./ c_s, 0, num_gridpts_theta, num_gridpts_theta) ...
584         - spdiags(u_sc .* (G_s * c_s) ./ (c_s .* c_s), 0, ...
585         num_gridpts_theta, num_gridpts_theta) ...
586         + spdiags(u_sc ./ c_s , 0, num_gridpts_theta, num_gridpts_theta) ...
587         * G_s ) * surf_extractor;
588
589     DH4_Dpsi_var = - ( spdiags(Dzeta_Dpsi .* (G_s * psi_s), 0, ...
590         num_gridpts_theta, num_gridpts_theta)...
591         + spdiags(zeta, 0, num_gridpts_theta, num_gridpts_theta) * G_s ) ...
592         * surf_extractor ...
593         + spdiags(Du_sc_Dzeta .* Dzeta_Dpsi .* (G_s * c_s) ./ c_s, 0, ...
594         num_gridpts_theta, num_gridpts_theta) * surf_extractor;
595
596     % -----
597     % 4: Build the complete Jacobian operator
598     % -----
599
600     J = [...
601         (DF1_Dc_const + DF1_Dc_var), ...
602             spalloc(num_gridpts_interior, num_gridpts, 0),...
603             (DF1_Dphi_const + DF1_Dphi_var), ...
604             (DF1_DU_const + DF1_DU_var) ; ...
605         (DF2_Dc_const + DF2_Dc_var), DF2_Dpsi_var , ...
606             spalloc(num_gridpts_interior, num_gridpts, 0), ...
607             spalloc(num_gridpts_interior, 1, 0); ...
608         ( DH1_Dc_const + DH1_Dc_var ), DH1_Dpsi_var, ...
609             spalloc(num_gridpts_theta, num_gridpts + 1, 0); ...
610         ( DH2_Dc_const + DH2_Dc_var ), DH2_Dpsi_var, ...
611             spalloc(num_gridpts_theta, num_gridpts + 1, 0);  ...
612         spalloc(num_gridpts_stokes, num_gridpts, 0), DF3_Dpsi_var, ...
613             DF3_Dphi_const, DF3_DU_const; ...
614             DM1_Dc_var, DM1_Dpsi_var, ...
615         spalloc(1, num_gridpts, 0), DM1_DU_const; ...
616             spalloc(num_gridpts_theta, num_gridpts, 0), ...
617             spalloc(num_gridpts_theta, num_gridpts, 0), DH3_Dphi_const, ...
618             spalloc(num_gridpts_theta, 1, 0); ...
```

```
619         DH4_Dc_var, DH4_Dpsi_var, DH4_Dphi_const, ...
620             spalloc(num_gridpts_theta, 1, 0) ...
621     ];
622
623     % ----
624     % 5: Update solution
625     % ----
626
627     delta_soln = -J\F;
628     c = c + delta_soln(1:num_gridpts);
629     psi = psi + delta_soln(num_gridpts + 1:2*num_gridpts);
630     phi = phi + delta_soln(2*num_gridpts + 1:3*num_gridpts);
631     U = U + delta_soln(3*num_gridpts + 1);
632
633     % ------------------------
634     % 6: Update Residual
635     % ------------------------
636
637     % extract surface concentration
638     c_s   = c(num_gridpts_r - 1:num_gridpts_r - 1:end);
639     psi_s = psi(num_gridpts_r - 1:num_gridpts_r - 1:end) - E * cos_theta;
640
641     % compute zeta potential
642     [V, zeta, q, w, Dzeta_Dpsi, Dzeta_Dc_s, Dq_Dpsi, Dq_Dc_s, Dw_Dpsi, ...
643         Dw_Dc_s, p1, p2, p3, p4, Dp1_Dpsi, Dp1_Dc_s, Dp2_Dpsi, Dp2_Dc_s, ...
644         Dp3_Dpsi, Dp3_Dc_s, Dp4_Dpsi, Dp4_Dc_s]...
645         = computeZetaPotential(psi_s, c_s, nu, Q);
646
647
648     % Chemiphoresis
649     [u_sc, u_sc_Dzeta] = chemiphoresis(0, zeta, nu);
650
651     % Bulk residual
652     F1 = F1_const_term + DF1_Dc_const*c - peclet * ( ...
653         (G_i{1}*c + c_infinity * ...
654         (G_inf{1} * ones(num_gridpts_theta, 1)) ) .* (P_i{1} * phi + ...
```

```
655         U * (P_inf{1} * ones(num_gridpts_theta, 1)) ) + ...
656         (G_i{2}*c + c_infinity * ...
657         (G_inf{2} * ones(num_gridpts_theta, 1)) ) .* (P_i{2} * phi + ...
658         U * (P_inf{2} * ones(num_gridpts_theta, 1)) ) ...
659         );
660
661     F2 = F2_const_term + DF2_Dc_const*c + D_i{1} * ...
662         (c .* (G_f{1} * psi)) + D_i{2} * (c .* (G_f{2} * psi));
663     F3 = DF3_Dphi_const*phi + DF3_DU_const*U + ...
664         (G_stokes{1} * L_f * psi) .* (G_stokes{2} * psi) - ...
665         (G_stokes{2} * L_f * psi) .* (G_stokes{1} * psi);
666
667     % Momentum conservation residual
668     Maxwell = (-1/3 * (1/2 * (G_i{1}*psi).^2 .* cos_theta_interior .* ...
669         (-9/4.*one_r_interior + 9/4.*one_r_interior.^3) ...
670         - 1/2 * (G_i{2}*psi).^2 .* cos_theta_interior .* ...
671         (-9/4.*one_r_interior + 9/4.*one_r_interior.^3) ...
672         + (G_i{1}*psi) .* (G_i{2}*psi) .* sin_theta_interior .* ...
673         (3/2.*one_r_interior.^3)) .* DrInt .* DthetaInt)' * OnesInt;
674     M1 = DM1_DU_const*U + Dtheta * ( zeta .* (G_s * psi_s) - ...
675         u_sc .* (G_s * c_s) ./ c_s ) + Maxwell;
676
677     % Boundary conditions residual
678     H1 = DH1_Dc_const * c + epsilon * D_s * ...
679         ((q + p3*peclet) ./ c_s .* (G_s * c_s) + ...
680         (w + p4*peclet) .* (G_s * psi_s)) - c_s .* (G_n_f * psi);
681     H2 = H2_const_term + DH2_Dc_const * c + epsilon * D_s * ...
682         ((w + p1*peclet) ./ c_s .* (G_s * c_s) + ...
683         (q + p2*peclet) .* (G_s * psi_s));
684     H3 = DH3_Dphi_const * phi;
685     H4 = DH4_Dphi_const * phi - zeta .* (G_s * psi_s)  + ...
686         u_sc .* (G_s * c_s) ./ c_s;
687
688     % Initialization
689     F = [F1; F2; H1; H2; F3; M1; H3; H4];
690     res = norm (F, inf)
```

```
691
692       % ------------------------
693       % 7: Update norms
694       % ------------------------
695
696       norm_delta_soln = norm(delta_soln, inf)
697       count = count + 1;
698
699       % show stats
700       if (show_stats > 1)
701           status = [res norm_delta_soln count]
702       end
703
704       % end Newton iteration loop
705       end
706
707       % update count_total
708       count_total = count_total + count;
709
710   % end loop over E_values
711   end
712
713
714   % ------------------------------------------------
715   % STEP 5: Calculate the velocity profile
716   % ------------------------------------------------
717
718   u_r = P_f{1} * phi + U * (P_inf_f{1} * ones(num_gridpts_theta, 1));
719   u_theta = P_f{2} * phi + U * (P_inf_f{2} * ones(num_gridpts_theta, 1));
720   u_s  = u_theta(num_gridpts_r - 1:num_gridpts_r - 1:end);
721   c_s  = c(num_gridpts_r - 1:num_gridpts_r - 1:end);
722   u_sc = - u_sc .* (G_s * c_s) ./ c_s;
723
724   % ------------------------------------------------
725   % STEP 6: append values at infinity to results
726   % ------------------------------------------------
```

```
727
728 c = finite_pt_extractor'*c;
729 c(1:num_gridpts_r:end) = c_infinity;
730 psi = finite_pt_extractor'*psi;
731 u_r = finite_pt_extractor'*u_r;
732 u_theta = finite_pt_extractor'*u_theta;
733 phi = finite_pt_extractor'*phi;
734
735
736 % ---------------------------------------------------
737 % STEP 7: error checking
738 % ---------------------------------------------------
739
740 % throw a warning if the solution has not converged
741 if (res > res_tol && norm_delta_soln > delta_tol)
742     mesg_id = 'solveHighFieldSteadyResponse3D:solutionNotConverged';
743     warning(mesg_id, ...
744     's3D:Solution NOT converged! res = %0.10f, norm_delta_soln = %0.10f', ...
745     res, norm_delta_soln);
746 end
747
748 % show computation statistics if requested
749 if (show_stats)
750     stats_string = sprintf( ...
751     '\n -s3D- \n Residual = %0.5g\n # iters = %d\n Last Change = %0.5g\n\n\n', ...
752     res, count_total, norm_delta_soln);
753     disp(stats_string);
754 end
```

A.5 Auxillary programs for dielectric particles (Steric model)

The main program used for dielectric particles is identical to the one in Section A.4.1 except that it calls computeZetaPotentialDiel.m instead of computeZetaPotential.m (lines 395 and 645). computeZetaPotentialDiel.m is almost identical to computeZe-

taPotential.m except for four main differences. Thus, we will omit the full program here for brevity.

Firstly, we add a line before the while loop after line 92:

```
1 zeta_guess = ones(length(Psi),1);
```

Secondly, all occurrences of `V - Psi` in the `while` loop are replaced by `zeta_guess`. Thirdly, in line 121, we replace `V = V + delta_V;` with the line

```
1 zeta_guess = zeta_guess + delta_V .* ones(length(zeta_guess),1);
```

Last but not least, after the `while` loop ends at line 136, we add `zeta = zeta_guess;`, and replace all subsequent instances of `V - Psi` by `zeta`.

A.6 Auxillary programs for dielectric decrement

A.6.1 Main program for dielectric decrement (full model)

The main program used for the dielectric decrement model is almost identical to the one in Section A.4.1 except for the following differences. Firstly, it has an additional argument `alpha` for the dielectric decrement parameter. Secondly, it calls DDcomputeZetaPotential.m instead of computeZetaPotential.m (lines 395 and 645). Thirdly, `res_tol` may have to be adjusted to make the computation time more reasonable. In addition, in lines 398 and 649, the following lines should be called instead:

```
1 [u_sc, Du_sc_Dzeta] = DDChemiphoresis(0, zeta, nu, c_s, alpha);
2 [u_el, Du_el_Dzeta] = DDElectroosmosis(0, zeta, nu, c_s, alpha);
```

Also, in `M1` and `H4`, `zeta` should be replaced by `u_el`. Furthermore, the Jacobian needs to be modified accordingly for the terms involving `M1` and `H4`:

```
1 DM1_Dc_var = Dtheta * (spdiags(Du_el_Dzeta .* Dzeta_Dc_s .* ...
```

```
2         (G_s * psi_s), 0, num_gridpts_theta, num_gridpts_theta) * ...
3         surf_extractor - ...
4         ( spdiags(Du_sc_Dzeta .* Dzeta_Dc_s .* (G_s * c_s) ./ c_s, 0, ...
5         num_gridpts_theta, num_gridpts_theta) ...
6         - spdiags(u_sc .* (G_s * c_s) ./ (c_s .* c_s), 0, ...
7         num_gridpts_theta, num_gridpts_theta) ...
8         + spdiags(u_sc ./ c_s , 0, num_gridpts_theta, num_gridpts_theta) ...
9         * G_s ) * surf_extractor ...
10        );
11 DM1_Dpsi_var = Dtheta * ( (spdiags(Du_el_Dzeta .* Dzeta_Dpsi .* ...
12        (G_s * psi_s), 0, num_gridpts_theta, num_gridpts_theta)...
13        + spdiags(u_el, 0, num_gridpts_theta, num_gridpts_theta) * G_s) ...
14        * surf_extractor + ...
15        spdiags(Du_sc_Dzeta .* Dzeta_Dpsi .* (G_s * c_s) ./ c_s, 0, ...
16        num_gridpts_theta, num_gridpts_theta) * surf_extractor ...
17        ) + DMaxwell_Dpsi;
18 DH4_Dc_var = spdiags(- Du_el_Dzeta .* Dzeta_Dc_s .* (G_s * psi_s), 0, ...
19        num_gridpts_theta, num_gridpts_theta) * surf_extractor + ...
20       ( spdiags(Du_sc_Dzeta .* Dzeta_Dc_s .* (G_s * c_s) ./ c_s, 0, ...
21       num_gridpts_theta, num_gridpts_theta) ...
22       - spdiags(u_sc .* (G_s * c_s) ./ (c_s .* c_s), 0, ...
23       num_gridpts_theta, num_gridpts_theta) ...
24       + spdiags(u_sc ./ c_s , 0, num_gridpts_theta, num_gridpts_theta) ...
25       * G_s ) * surf_extractor;
26 DH4_Dpsi_var = - ( spdiags(Du_el_Dzeta .* Dzeta_Dpsi .* (G_s * psi_s), ...
27        0, num_gridpts_theta, num_gridpts_theta)...
28        + spdiags(u_el, 0, num_gridpts_theta, num_gridpts_theta) * G_s ) ...
29    * surf_extractor ...
30    + spdiags(Du_sc_Dzeta .* Dzeta_Dpsi .* (G_s * c_s) ./ c_s, 0, ...
31    num_gridpts_theta, num_gridpts_theta) * surf_extractor;
```

A.6.2 Main program for dielectric decrement (simplified model)

For the simplified model without surface conduction, everything in the the original main program in Section A.4.1 after Step 3 (line 298) can be omitted and replaced

with the following:

```
if (generate_initial_guess)
    U = 0;
    V = 0;
end

c = c_infinity * ones(num_gridpts, 1);
psi = - E * cos_theta_full .* (...
    finite_pt_extractor*kron(ones(N_theta, 1), 1/2./r.^2));
%add E r cos(theta) to known solution to get psi - psi is the deviation
%from the general solution without a disturbing charge

% Extract surface concentration and potential
c_s = c(num_gridpts_r - 1:num_gridpts_r - 1:end);
psi_s = psi(num_gridpts_r - 1:num_gridpts_r - 1:end) - E * cos_theta;

% Compute zeta potential
[V, zeta] = DDcomputeZetaPotentialPSsimp(psi_s, c_s, alpha, nu, Q, V);

Maxwell = (-1/3 * (1/2 * (G_i{1}*psi).^2 .* cos_theta_interior .* ...
    (-9/4.*one_r_interior + 9/4.*one_r_interior.^3) ...
    - 1/2 * (G_i{2}*psi).^2 .* cos_theta_interior .* ...
    (-9/4.*one_r_interior + 9/4.*one_r_interior.^3) ...
    + (G_i{1}*psi) .* (G_i{2}*psi) .* sin_theta_interior .* ...
    (3/2.*one_r_interior.^3)) .* DrInt .* DthetaInt)' * OnesInt;

u_s = zeta .* (G_s * psi_s);
U = 0.5 * Dtheta * u_s + 0.5 * Maxwell;
```

In addition, all unnecessary operators in the preamble can be removed to reduce computation time. Also, a number of output variables in DDcomputeZetaPotential.m are no longer necessary in the simplified model.

A.6.3 Calculate diffusio-osmotic velocity: DDChemiphoresis.m

```
% DDChemiphoresis computes the diffusio-osmotic velocity at the surface of
% the particle in the dielectric decrement case.
%
% function [u_cp, Du_cp_Dzeta] = DDChemiphoresis(xi, zeta, nu, c_s, alpha, res_tol)
%
% Inputs:
% xi (required):        Lower limit of integration
% zeta (required):      Zeta potential at the surface of the particle
% nu (required):        Steric parameter
% c_s (required):       Ionic concentration in the bulk solution at the
%       boundary of the particle
% alpha (required):     Dielectric decrement parameter
% res_tol (optional):   Tolerance for residual in computation of w
%
% Outputs:
% u_cp:         Diffusio-osmotic velocity
% Du_cp_Dzeta:  Derivative of the diffusio-osmotic velocity with respect to
%   the zeta potential
% ------------------------------------------------------------------------

function [u_cp, Du_cp_Dzeta] = DDChemiphoresis(xi, zeta, nu, c_s, alpha, res_tol)

% ------------------------------------------------------------
% STEP 1: check argument list and set default values
% ------------------------------------------------------------

max_args = 6;
if (nargin < 5)
    error('MATLAB:missingArgs','DDChemiphoresis:missing arguments');
end
if (nargin < 6)
```

```
32      res_tol = 1E-6;
33  end
34
35  % ------------------------------------------------------------------
36  % STEP 2: Initialization
37  % ------------------------------------------------------------------
38  num_gridpts = length(zeta);
39
40  % ------------------------------------------------------------------
41  % STEP 3: Sort the values of the zeta potential by ascending order
42  % ------------------------------------------------------------------
43
44  matrixtosort = [abs(zeta) zeta c_s sign(zeta)];
45  [matrixsorted, idx] = sortrows(matrixtosort,1);
46  zeta_s = matrixsorted(:,2);
47  c_s_s = matrixsorted(:,3);
48  zeta_sign_s = matrixsorted(:,4);
49
50  % ------------------------------------------------------------------
51  % STEP 4: Perform the double integration
52  % ------------------------------------------------------------------
53
54  u_cp = spalloc(num_gridpts, 1, 0);
55  Du_cp_Dzeta = spalloc(num_gridpts, 1, 0);
56
57  w_min_positive = xi + eps;
58  w_min_negative = xi - eps;
59  w_min = 0.0;
60  res_w_positive = 0.0;
61  res_w_negative = 0.0;
62  res_w = 0.0;
63
64  %for zeta profile
65  zeta_prev_positive = 0;
66  zeta_prev_negative = 0;
67  zeta_prev = 0;
```

```
68 zeta_length = 500; %Vary this for closer convergence to actual result
69
70 for n = 1:num_gridpts
71
72     % Choose sign
73     if (zeta_sign_s(n) == 1)
74         w_min = w_min_positive;
75         res_w = res_w_positive;
76         zeta_prev = zeta_prev_positive;
77     else
78         w_min = w_min_negative;
79         res_w = res_w_negative;
80         zeta_prev = zeta_prev_negative;
81     end
82
83     % Initialization
84     w_max = zeta_s(n);
85
86     % Generate zeta profile
87     [y,zeta_vector] = DDzetaProfileGuess(w_max,alpha,nu,zeta_length,5,...
88         c_s_s(n),zeta_prev);
89     if (zeta_sign_s(n) == 1)
90         zeta_prev_positive = zeta_vector;
91     else
92         zeta_prev_negative = zeta_vector;
93     end
94     zeta_vector = zeta_vector(zeta_vector~=0);
95     zeta_vector_length = length(zeta_vector);
96     Dzeta_vector_Dy = zeros(zeta_vector_length,1);
97     for j = 2:zeta_vector_length
98         Dzeta_vector_Dy(j) = -(zeta_vector(j-1)-zeta_vector(j)) / ...
99             (y(j)-y(j-1));
100    end
101
102    % Evaluate the integral
103    res_w = res_w + triplequad(@tripfun, 1E-12, 1, 1E-12, 1, w_min, ...
```

```
104         w_max, res_tol);
105
106     % Derivative w.r.t the zeta potential
107     fun = @(u,v) tripfun(u, v, w_max);
108     Du_cp_Dzeta(n) = dblquad(fun, 1E-12, 1, 1E-12, 1, res_tol);
109
110     % Update
111     u_cp(n) = res_w;
112
113     % Choose sign
114     if (zeta_sign_s(n) == 1)
115         w_min_positive = w_max;
116         res_w_positive = res_w;
117     else
118         w_min_negative = w_max;
119         res_w_negative = res_w;
120     end
121
122 end
123
124 % ------------------------------------------------------------
125 % STEP 5: Re-sort the arrays values
126 % ------------------------------------------------------------
127
128 u_cp(idx) = u_cp;
129 Du_cp_Dzeta(idx) = Du_cp_Dzeta;
130
131 % ------------------------------------------------------------
132 % Sub-function: Define the function to integrate
133 % ------------------------------------------------------------
134
135
136 function result = tripfun(u,v,w)
137     gradient = interp1(zeta_vector,Dzeta_vector_Dy,u.*v.*w);
138     gradient_vw = interp1(zeta_vector,Dzeta_vector_Dy,v.*w);
139     gradient_w = interp1(zeta_vector,Dzeta_vector_Dy,w);
```

```
140     if (gradient_vw == 0)
141         gradient_vw = eps;
142     end
143     if (gradient_w == 0)
144         gradient_w = eps;
145     end
146     S = sinh(u .* v .* w);
147     C = cosh(u .* v .* w);
148     E = exp(-alpha * gradient.^2);
149     result = S .* E ./ (1 + nu * (C .* E - 1)) .* v .* w.^2 ./ ...
150         gradient_vw ./ gradient_w;
151 end
152
153 end
```

A.6.4 Calculate electro-osmotic velocity: DDElectroosmosis.m

```
1  % DDElectroosmosis computes the electro-osmotic velocity at the surface of
2  % the particle in the dielectric decrement case.
3  %
4  % function [u_el, Du_el_Dzeta] = DDElectroosmosis(xi, zeta, nu, c_s, alpha, res_tol)
5  %
6  % Inputs:
7  % xi (required):        Lower limit of integration
8  % zeta (required):      Zeta potential at the surface of the particle
9  % nu (required):        Steric parameter
10 % c_s (required):       Ionic concentration in the bulk solution at the
11 %       boundary of the particle
12 % alpha (required):     Dielectric decrement parameter
13 % res_tol (optional):   Tolerance for residual in computation of w
14 %
15 % Outputs:
16 % u_el:         Electro-osmotic velocity
```

```
% Du_el_Dzeta:  Derivative of the electro-osmotic velocity with respect to
%   the zeta potential
% ------------------------------------------------------------------------------

function [u_el, Du_el_Dzeta] = DDElectroosmosis(xi, zeta, nu, c_s, alpha, res_tol)

% ------------------------------------------------------------
% STEP 1: check argument list and set default values
% ------------------------------------------------------------

max_args = 6;
if (nargin < 5)
    error('MATLAB:missingArgs','DDElectroosmosis:missing arguments');
end
if (nargin < 6)
    res_tol = 1E-6;
end

% ------------------------------------------------------------
% STEP 2: Initialization
% ------------------------------------------------------------
num_gridpts = length(zeta);

% ------------------------------------------------------------
% STEP 3: Sort the values of the zeta potential by ascending order
% ------------------------------------------------------------

matrixtosort = [abs(zeta) zeta c_s sign(zeta)];
[matrixsorted, idx] = sortrows(matrixtosort,1);
zeta_s = matrixsorted(:,2);
c_s_s = matrixsorted(:,3);
zeta_sign_s = matrixsorted(:,4);

% ------------------------------------------------------------
% STEP 4: Perform the double integration
% ------------------------------------------------------------
```

```
53
54 u_el = spalloc(num_gridpts, 1, 0);
55 Du_el_Dzeta = spalloc(num_gridpts, 1, 0);
56
57 w_min_positive = xi + eps;
58 w_min_negative = xi - eps;
59 w_min = 0.0;
60 res_w_positive = 0.0;
61 res_w_negative = 0.0;
62 res_w = 0.0;
63
64 %for zeta profile
65 zeta_prev_positive = 0;
66 zeta_prev_negative = 0;
67 zeta_prev = 0;
68 zeta_length = 500; %Vary this for closer convergence to actual result
69
70 for n = 1:num_gridpts
71
72     % Choose sign
73     if (zeta_sign_s(n) == 1)
74         w_min = w_min_positive;
75         res_w = res_w_positive;
76         zeta_prev = zeta_prev_positive;
77     else
78         w_min = w_min_negative;
79         res_w = res_w_negative;
80         zeta_prev = zeta_prev_negative;
81     end
82
83     % Initialization
84     w_max = zeta_s(n);
85
86     % Generate zeta profile
87     [y,zeta_vector] = DDzetaProfileGuess(w_max,alpha,nu,zeta_length,5,...
88         c_s_s(n),zeta_prev);
```

```
89      if (zeta_sign_s(n) == 1)
90          zeta_prev_positive = zeta_vector;
91      else
92          zeta_prev_negative = zeta_vector;
93      end
94      %zeta_prev = zeta_vector;
95      zeta_vector = zeta_vector(zeta_vector~=0);
96      zeta_vector_length = length(zeta_vector);
97      Dzeta_vector_Dy = zeros(zeta_vector_length,1);
98      for j = 2:zeta_vector_length
99          Dzeta_vector_Dy(j) = -(zeta_vector(j-1)-zeta_vector(j)) / ...
100             (y(j)-y(j-1));
101     end
102
103     % Evaluate the integral
104     res_w = res_w + triplequad(@tripfun, 1E-12, 1, 1E-12, 1, w_min, ...
105         w_max, res_tol);
106
107     % Derivative w.r.t the zeta potential
108     fun = @(u,v) tripfun(u, v, w_max);
109     Du_el_Dzeta(n) = dblquad(fun, 1E-12, 1, 1E-12, 1, res_tol);
110
111     % Update
112     u_el(n) = res_w;
113
114     % Choose sign
115     if (zeta_sign_s(n) == 1)
116         w_min_positive = w_max;
117         res_w_positive = res_w;
118     else
119         w_min_negative = w_max;
120         res_w_negative = res_w;
121     end
122
123 end
124
```

```
125 % ------------------------------------------------------------
126 % STEP 5: Re-sort the arrays values
127 % ------------------------------------------------------------
128
129 u_el(idx) = u_el;
130 Du_el_Dzeta(idx) = Du_el_Dzeta;
131
132 % ------------------------------------------------------------
133 % Sub-function: Define the function to integrate
134 % ------------------------------------------------------------
135
136
137 function result = tripfun(u,v,w)
138     gradient = interp1(zeta_vector,Dzeta_vector_Dy,u.*v.*w);
139     gradient_vw = interp1(zeta_vector,Dzeta_vector_Dy,v.*w);
140     gradient_w = interp1(zeta_vector,Dzeta_vector_Dy,w);
141     if (gradient_vw == 0)
142         gradient_vw = eps;
143     end
144     if (gradient_w == 0)
145         gradient_w = eps;
146     end
147     S = sinh(u .* v .* w);
148     C = cosh(u .* v .* w);
149     E = exp(-alpha * gradient.^2);
150     result = (C .* E ./ (1 + nu * (C .* E - 1)) - nu * S.^2 .* E.^2 ./ ...
151         (1 + nu * (C .* E - 1)).^2) .* v .* w.^2 ./ gradient_vw ./ gradient_w;
152 end
153
154 end
```

A.6.5 Solve for zeta potential distribution around particle: DDcomputeZetaPotential.m

```
% DDcomputeZetaPotential computes the zeta potential on the surface of the
% particle using the charge conservation of the particle for the dielectric
% decrement case.
%
%function [V, zeta, q, w, Dzeta_Dpsi, Dzeta_Dc_s, Dq_Dpsi, Dq_Dc_s, ...
%    Dw_Dpsi, Dw_Dc_s, p1, p2, p3, p4, Dp1_Dpsi, Dp1_Dc_s, Dp2_Dpsi, ...
%    Dp2_Dc_s, Dp3_Dpsi, Dp3_Dc_s, Dp4_Dpsi, Dp4_Dc_s...
%    ] = DDcomputeZetaPotential( ...
%    Psi, c_s, alpha, nu, Q, V_init, ... % physical parameters
%    res_tol, delta_zeta_tol, max_iters, ... % iteration parameters
%    show_stats ... % show statistics flag
%   )
%
% Inputs:
% Psi (required):               Surface potential
% c_s (required):               Surface concentration
% alpha (required):             Dielectric decrement parameter
% nu (required):                Packing parameter
% Q (required):                 Total surface charge of the particle
% V_init (optional):            Initial guess for the potential of the particle
% res_tol (optional):           Tolerance for residual in computation of zeta
% delta_zeta_tol (optional):    Tolerance for change in zeta
% max_iters (optional):         Maximum number of Newton iterations
% show_stats (optional):        Show statistics if set to a non-zero value
%
% Outputs:
% V:            Potential of the particle
% zeta:         Zeta potential at each point of the surface
% q:            Excess charge
% w:            Excess concentration
% Dzeta_Dpsi:   Derivative of zeta with respect to the bulk potential
% Dzeta_Dc_s:   Derivative of zeta with respect to the surface concentration
% Dq_Dpsi:      Derivative of q with respect to the bulk potential
% Dq_Dc_s:      Derivative of q with respect to the surface concentration
% Dw_Dpsi:      Derivative of w with respect to the bulk potential
% Dw_Dc_s:      Derivative of w with respect to the surface concentration
```

```
37 % Note that p_i are quantities necessary to take into account
38 % consideration, and the relevant derivatives with respect to the bulk
39 % potential and the surface concentration are also outputs.
40 % ------------------------------------------------------------------------------
41
42 function [V, zeta, q, w, Dzeta_Dpsi, Dzeta_Dc_s, Dq_Dpsi, Dq_Dc_s, ...
43     Dw_Dpsi, Dw_Dc_s, p1, p2, p3, p4, Dp1_Dpsi, Dp1_Dc_s, Dp2_Dpsi, ...
44     Dp2_Dc_s, Dp3_Dpsi, Dp3_Dc_s, Dp4_Dpsi, Dp4_Dc_s...
45     ] = DDcomputeZetaPotential( ...
46     Psi, c_s, alpha, nu, Q, V_init, ... % physical parameters
47     res_tol, delta_zeta_tol, max_iters, ... % iteration parameters
48     show_stats ... % show statistics flag
49     )
50
51 % ------------------------------------------------------------
52 % STEP 1: check argument list and set default values
53 % ------------------------------------------------------------
54
55 max_args = 10;
56 if (nargin < 5)
57     error('MATLAB:missingArgs','DDcomputeZetaPotential:missing arguments');
58 end
59 if (nargin < max_args)
60     show_stats = 0;
61 end
62 if (nargin < max_args - 1)
63     max_iters = 100;
64 end
65 if (nargin < max_args - 2)
66     delta_zeta_tol = 1E-13;
67 end
68 if (nargin < max_args - 3)
69     res_tol = 1E-7;
70 end
71 if (nargin < max_args - 4)
72     V_init = 0.0;
```

```
73 end
74
75 % ------------------------------------------------------------
76 % STEP 2: Define the integration operator
77 % ------------------------------------------------------------
78
79 N_theta = length(Psi);
80 [~, theta, DeltaTheta] = DM_cosine_interior(N_theta);
81
82 % ------------------------------------------------------------
83 % STEP 3: Compute the initial residual
84 % ------------------------------------------------------------
85
86 % Initialize iteration
87 V = V_init;
88 delta_V = 1;
89 res = 1;
90 norm_res = norm(res, inf);
91 norm_delta_zeta = norm(delta_V, inf);
92 count = 0;
93
94 zeta_guess = V - Psi;
95
96 [q,Dq_Dzeta,~,~,~,~,~,~,~,~,~,~] = DDgenerateQty(zeta_guess,alpha,nu,c_s,0);
97 res = Q + DeltaTheta * (q .* sin(theta));
98
99
100 % ------------------------------------------------------------
101 % STEP 4: Newton iteration
102 % ------------------------------------------------------------
103
104 while (norm_res > res_tol && norm_delta_zeta > delta_zeta_tol && count < max_iters)
105
106     J = DeltaTheta * (Dq_Dzeta .* sin(theta));
107
108     delta_V = - res/J;
```

```
109     V = V + delta_V;
110
111     zeta_guess = V - Psi;
112
113     [q,Dq_Dzeta,~,~,~,~,~,~,~,~,~,~] = ...
114         DDgenerateQty(zeta_guess,alpha,nu,c_s,0);
115     res = Q + DeltaTheta * (q .* sin(theta));
116
117     % Update norm_res, norm_delta_zeta and count
118     norm_res = norm(res, inf);
119     norm_delta_zeta = norm(delta_V, inf);
120     count = count + 1;
121
122 end
123
124 % ------------------------------------------------------------
125 % STEP 5: Output
126 % ------------------------------------------------------------
127
128 zeta = V - Psi;
129 q = 2*q;
130 Dq_Dzeta = 2*Dq_Dzeta;
131
132 [~,~,w,Dw_Dzeta,p1,p2,p3,p4,Dp1_Dzeta,Dp2_Dzeta,Dp3_Dzeta,Dp4_Dzeta...
133     ] = DDgenerateQty(zeta,alpha,nu,c_s,1)
134
135 Dzeta_Dpsi = 0;
136 % by assumption; V must match psi to keep Q constant
137 % (i.e. DV_Dpsi = 1) and Dzeta_Dpsi = DV_Dpsi - Dpsi_Dpsi
138 Dzeta_Dc_s = -0.5*q./c_s./Dq_Dzeta;
139 % Dq_Dc_s = Dq_Dc_s|zeta + Dq_Dzeta * Dzeta_Dc_s = 0 ;
140 % Dq_Dc_s|zeta = 0.5 * q / c_s
141
142 Dq_Dpsi = Dq_Dzeta .* Dzeta_Dpsi; % 0
143 Dq_Dc_s = 0; % by assumption
144
```

```
145 Dw_Dpsi = Dw_Dzeta .* Dzeta_Dpsi; % 0
146 Dw_Dc_s = 0.5*w./c_s + Dw_Dzeta.*Dzeta_Dc_s;
147
148 w = 2*w;
149 Dw_Dc_s = 2*Dw_Dc_s;
150
151 Dp1_Dpsi = Dp1_Dzeta .* Dzeta_Dpsi; % 0
152 Dp1_Dc_s = 0.5*p1./c_s + Dp1_Dzeta.*Dzeta_Dc_s;
153
154 Dp2_Dpsi = Dp2_Dzeta .* Dzeta_Dpsi; % 0
155 Dp2_Dc_s = 0.5*p2./c_s + Dp2_Dzeta.*Dzeta_Dc_s;
156
157 Dp3_Dpsi = Dp3_Dzeta .* Dzeta_Dpsi; % 0
158 Dp3_Dc_s = 0.5*p3./c_s + Dp3_Dzeta.*Dzeta_Dc_s;
159
160 Dp4_Dpsi = Dp4_Dzeta .* Dzeta_Dpsi; % 0
161 Dp4_Dc_s = 0.5*p4./c_s + Dp4_Dzeta.*Dzeta_Dc_s;
162
163
164 % ------------------------------------------------------------
165 % STEP 6: Error checking
166 % ------------------------------------------------------------
167
168 % Throw a warning if the solution has not converged
169 if (norm_res > res_tol && norm_delta_zeta > delta_zeta_tol)
170     mesg_id = 'DDcomputeZetaPotential:solutionNotConverged';
171     warning(mesg_id, 'DDcZP:NOT converged! res = %0.10f, delta_zeta = %0.10f', ...
172         norm_res, norm_delta_zeta);
173 end
174
175 % ------------------------------------------------------------
176 % STEP 7: Show computation statistics if requested
177 % ------------------------------------------------------------
178 if (show_stats)
179     stats_string = sprintf( ...
180         '\n -DDcZP- \n Res = %0.5g\n # iters = %d\n Change = %0.5g\n\n\n', ...
```

```
181         norm_res, count, norm_delta_zeta);
182     disp(stats_string);
183 end
```

A.6.6 Helper function to generate electric potential profile in EDL: DDgenerateQty.m

```
1  % DDgenerateQty is a helper function for DDcomputeZetaPotential.
2  %
3  % function [q,Dq_Dzeta,w,Dw_Dzeta,p1,p2,p3,p4,Dp1_Dzeta,Dp2_Dzeta,Dp3_Dzeta,...
4  %     Dp4_Dzeta] = DDgenerateQty(zeta_guess,alpha,nu,c_s,phase)
5  %
6  % Inputs:
7  % zeta_guess (required):        Guess for zeta potential distribution
8  % alpha (required):             Dielectric decrement parameter
9  % nu (required):                Packing parameter
10 % c_s (required):                Surface concentration
11 % phase:                        Determines outputs
12 % phase 0: q, Dq_Dzeta
13 % phase 1: w, Dw_Dzeta and p_i + related variables
14 %
15 % Outputs:
16 % q:            Excess charge
17 % w:            Excess concentration
18 % Dq_Dzeta:     Derivative of q with respect to the zeta potential
19 % Dw_Dzeta:     Derivative of w with respect to the zeta potential
20 % p_i and its relevant derivatives are required if advection is taken into
21 % consideration (Pe =/= 0)
22 % ------------------------------------------------------------------------
23
24 function [q,Dq_Dzeta,w,Dw_Dzeta,p1,p2,p3,p4,Dp1_Dzeta,Dp2_Dzeta,Dp3_Dzeta,...
25     Dp4_Dzeta] = DDgenerateQty(zeta_guess,alpha,nu,c_s,phase)
26
27 N_theta = length(zeta_guess);
```

```
28
29 % Sort zeta_guess in ascending absolute value order
30 matrixtosort = [zeta_guess abs(zeta_guess) c_s];
31 [matrixsorted, idx] = sortrows(matrixtosort,2);
32 zeta_sort = matrixsorted(:,1);
33 c_s_sort = matrixsorted(:,3);
34
35 % Initialize variables
36 zeta_prev = 0;
37 zeta_length = 1500; %Vary this for closer convergence to actual result
38 q = zeros(N_theta,1);
39 w = zeros(N_theta,1);
40 p1 = zeros(N_theta,1);
41 p2 = zeros(N_theta,1);
42 p3 = zeros(N_theta,1);
43 p4 = zeros(N_theta,1);
44 Dq_Dzeta = zeros(N_theta,1);
45 Dw_Dzeta = zeros(N_theta,1);
46 Dp1_Dzeta = zeros(N_theta,1);
47 Dp2_Dzeta = zeros(N_theta,1);
48 Dp3_Dzeta = zeros(N_theta,1);
49 Dp4_Dzeta = zeros(N_theta,1);
50
51 % Calculate quantities
52 for i = 1:N_theta
53     [y,zeta] = DDzetaProfileGuess(zeta_sort(i),alpha,nu,zeta_length,...
54         2000,c_s_sort(i),zeta_prev);
55     y = y./1E3; %Scaling in DDzetaProfileGuess
56     Dzeta_Dy = zeros(zeta_length,1);
57     if (phase == 1)
58         KD1 = zeros(zeta_length,1);
59         KD1a = zeros(zeta_length,1);
60         KD2 = zeros(zeta_length,1);
61         KD2a = zeros(zeta_length,1);
62         [KD1(1),KD1a(1)] = DDElectroosmosis(zeta(1),zeta(end),nu,c_s,alpha);
63         [KD2(1),KD2a(1)] = DDChemiphoresis(zeta(1),zeta(end),nu,c_s,alpha);
```

```
    end
    for j = 2:zeta_length
        Dzeta_Dy(j) = -(zeta(j-1)-zeta(j)) / (y(j)-y(j-1));
        if (phase == 1)
            [KD1(j),KD1a(j)] = DDElectroosmosis(zeta(j),...
                zeta(end),nu,c_s,alpha);
            [KD2(j),KD2a(j)] = DDChemiphoresis(zeta(j),...
                zeta(end),nu,c_s,alpha);
        end
    end
    exp_alpha = exp(-alpha*Dzeta_Dy.^2);
    S = sinh(zeta).*exp_alpha;
    C = cosh(zeta).*exp_alpha;
    D = 1 + nu*(C-1);
    rho = c_s_sort(i)*S ./ D;
    if (phase == 0)
        q(i) = trapz(y,rho);
        Dq_Dzeta(i) = rho(end)/Dzeta_Dy(end);
        %Dq_Dzeta = Dq_Dy|y=0 / Dzeta_Dy|y=0
    elseif (phase == 1)
        exc = C ./ D;
        EP = exp(zeta).*exp_alpha;
        EM = exp(zeta).*exp_alpha;
        cp = EM ./ D;
        cm = EP ./ D;
        w(i) = -c_s_sort(i)*trapz(y,exc-1);
        Dw_Dzeta(i) = -c_s_sort(i)*(exc(end)-1)/Dzeta_Dy(end);
        %Dw_Dzeta = Dw_Dy|y=0 / Dzeta_Dy|y=0
        p1(i) = -c_s_sort(i)*trapz(y,((cp+cm).*KD2-2*KD2(1)));
        Dp1_Dzeta(i) = -c_s_sort(i)*...
            (-2*KD2(1)+trapz(y,((cp+cm).*KD2a-KD2a(1))));
        p2(i) = c_s_sort(i)*trapz(y,((cp+cm).*KD1-2*KD1(1)));
        Dp2_Dzeta(i) = c_s_sort(i)*...
            (-2*KD1(1)+trapz(y,((cp+cm).*KD1a-KD1a(1))));
        p3(i) = -c_s_sort(i)*trapz(y,((cp-cm).*KD2));
        Dp3_Dzeta(i) = -c_s_sort(i)*trapz(y,((cp-cm).*KD2a));
```

```
100         p4(i) = c_s_sort(i)*trapz(y,((cp-cm).*KD1));
101         Dp4_Dzeta(i) = c_s_sort(i)*trapz(y,((cp-cm).*KD1a));
102     end
103     zeta_prev = zeta;
104 end
105
106 if (phase == 0)
107     q(idx) = q;
108     Dq_Dzeta(idx) = Dq_Dzeta;
109 elseif (phase == 1)
110     w(idx) = w;
111     Dw_Dzeta(idx) = Dw_Dzeta;
112     p1(idx) = p1;
113     p2(idx) = p2;
114     p3(idx) = p3;
115     p4(idx) = p4;
116     Dp1_Dzeta(idx) = Dp1_Dzeta;
117     Dp2_Dzeta(idx) = Dp2_Dzeta;
118     Dp3_Dzeta(idx) = Dp3_Dzeta;
119     Dp4_Dzeta(idx) = Dp4_Dzeta;
120 end
121
122 end
```

A.6.7 Generate electric potential profile in EDL: DDzetaProfileGuess.m

```
1 % DDzetaProfileGuess solves for the electric potential profile in the
2 % double layer using the Poisson-Boltzmann equation modified for steric and
3 % dielectric decrement effects.
4 %
5 % function [z, phi] = DDzetaProfileGuess(zeta, delta_c, nu, N_z, L_z, c_s, ...
6 % guess, res_tol, delta_zeta_tol, max_iters, show_stats)
7 %
```

```
8  % Inputs:
9  % zeta (required):              Zeta potential at the surface of the particle
10 % delta_c (required):           Dielectric decrement parameter
11 % nu (required):                Steric parameter
12 % N_z (required):               Grid size
13 % L_z (required):               Grid scale factor
14 % c_s (required):       Ionic concentration in the bulk solution at the
15 %       boundary of the particle
16 % guess (optional):             Guess for the potential profile
17 % res_tol (optional):           Tolerance for residual in computation of zeta
18 % delta_zeta_tol (optional):    Tolerance for change in zeta
19 % max_iters (optional):         Maximum number of Newton iterations
20 % show_stats (optional):        Show statistics if set to a non-zero value
21 %
22 % Output:
23 % z:        Grid points
24 % phi:      Corresponding electric potential
25 % ------------------------------------------------------------------------
26
27 function [z, phi] = DDzetaProfileGuess(zeta, delta_c, nu, N_z, L_z, c_s, ...
28     guess, res_tol, delta_zeta_tol, max_iters, show_stats)
29
30
31 % ------------------------------------------------------------
32 % STEP 1: check argument list and set default values
33 % ------------------------------------------------------------
34
35 max_args = 11;
36 if (nargin < 7)
37     error('MATLAB:missingArgs','DDzetaProfileGuess:missing arguments');
38 end
39 if (nargin < max_args)
40     show_stats = 0;
41 end
42 if (nargin < max_args - 1)
43     max_iters = 500;
```

```
44 end
45 if (nargin < max_args - 2)
46     delta_norm_tol = 1E-7;
47 end
48 if (nargin < max_args - 3)
49     res_tol = 1E-4;
50 end
51
52 % ------------------------------------------------------------------------
53 % STEP 2: Initialization
54 % ------------------------------------------------------------------------
55
56 num_gridpts_theta = length(zeta);
57
58 % Derivative operator
59 [D_z, z] = DM_TL(N_z, L_z);
60
61 % Extraction operators
62 finite_pt_extractor = spdiags(ones(N_z, 1), 1, N_z, N_z + 1);
63 interior_extractor = spdiags(ones(N_z - 1, 1), 1, N_z - 1, N_z + 1);
64
65 % Interior and finite points operators
66 D_f = finite_pt_extractor  * D_z * finite_pt_extractor';
67 D_i = interior_extractor  * D_z * finite_pt_extractor';
68
69 % ------------------------------------------------------------------------
70 % STEP 3: Calculates the charge
71 % ------------------------------------------------------------------------
72
73 %scaling factors to improve convergence
74 Ds = 1E3;
75 D2s = Ds.^2;
76
77 for p = (1:num_gridpts_theta)
78
79     if (delta_c ~= 0)
```

```
80          alpha_store = delta_c;
81          delta_c = 0;
82          alpha_max = ceil(alpha_store/0.01) + 2;
83          % The step size may have to be decreased to improve convergence
84      else
85          alpha_store = delta_c;
86          alpha_max = 2;
87      end
88
89      % First estimate of the solution
90      if (length(guess) ~= N_z)
91          phi = zeta(p) * exp(-z(2:end));
92      else
93          phi = guess / guess(end) * zeta(p);
94      end
95
96      alpha_count = 1;
97
98      while (alpha_count < alpha_max)
99
100     % ------------------------------------------------------------------------
101     % 1: Calculates the initial residual
102     % ------------------------------------------------------------------------
103
104     % Constant part of the equation
105     F2_const = - zeta(p);
106
107     % Constant part of the Jacobian
108     DF1_const = D2s * D_i * D_f;
109     DF2_const = spalloc(1, N_z, 0);
110     DF2_const(end) = 1.0;
111
112     % Excess ion concentration, Local charge
113     f = exp( - delta_c * (D_f * phi).^2).^D2s;
114     rho = c_s(p) * sinh(phi) .* f ./ (1 - nu + nu * f .* cosh(phi) );
115     exc = c_s(p) * cosh(phi) .* f ./ (1 - nu + nu * f .* cosh(phi) );
```

```
116
117     % Calculates the first residual
118     F1 = DF1_const * phi - 2.0 * delta_c * D2s * D_i * (spdiags(exc, 0, ...
119         N_z, N_z) * D_f * phi) - rho(1:end - 1);
120     F2 = F2_const + DF2_const * phi;
121     F = [F1; F2];
122     norm_res = norm(F, inf);
123
124     % Calculates the first residual
125     norm_delta_soln = 1.0;
126     count = 0;
127
128     % ------------------------------------------------------------------------
129     % 2: Newton's iteration
130     % ------------------------------------------------------------------------
131
132     while (norm_res > res_tol && norm_delta_soln > delta_norm_tol && ...
133             count < max_iters)
134
135         % 1: Estimate quantities used to calculate the Jacobian
136
137         f = exp( - delta_c * (D_f * phi).^2) .^ D2s;
138         Df_Dphi = - 2.0 * delta_c * f * Ds .* (D_f * phi);
139
140         u = sinh(phi) .* f;
141         v = cosh(phi) .* f;
142         w = 1 - nu + nu * v;
143
144         Du_Dphi =  cosh(phi) .* f;
145         Dv_Dphi =  sinh(phi) .* f;
146         Dw_Dphi =  nu * sinh(phi) .* f;
147
148         Du_Dphi_grad =  sinh(phi) .* Df_Dphi;
149         Dv_Dphi_grad =  cosh(phi) .* Df_Dphi;
150         Dw_Dphi_grad =  nu * cosh(phi) .* Df_Dphi;
151
```

```
152
153         % 2: Differential of the excess ionic concentration operator
154         exc = c_s(p) * v ./ w;
155         Dexc_Dphi = - spdiags( c_s(p) * (Dv_Dphi .* w - v .* Dw_Dphi) ./ ...
156             (w.^2) .* (Ds .* D_f * phi), 0, N_z, N_z) - ...
157             spdiags( c_s(p) * (Dv_Dphi_grad .* w - v .* Dw_Dphi_grad) ./ ...
158             (w.^2) .* (D_f * phi), 0, N_z, N_z) * D2s * D_f;
159         Drho_Dphi = - spdiags( c_s(p) * (Du_Dphi .* w - u .* Dw_Dphi) ./ ...
160             (w.^2), 0, N_z - 1, N_z ) - ...
161             spdiags( c_s(p) * (Du_Dphi_grad .* w - u .* Dw_Dphi_grad) ./...
162             (w.^2), 0, N_z - 1, N_z - 1) * Ds * D_i;
163
164
165         % Jacobian of the first operator
166         DF1_var1 =  2.0 * delta_c * Ds * D_i * (Dexc_Dphi - ...
167             spdiags(exc, 0, N_z, N_z) .* Ds * D_f);
168
169
170         % Jacobian of the second operator
171         DF1_var2 = Drho_Dphi;
172
173
174         % Jacobian matrix
175         J = [DF1_const + DF1_var1 + DF1_var2; DF2_const];
176
177         % Update solution
178         delta_soln = - J\F;
179         phi = phi + delta_soln;
180
181
182         % Excess ion concentration, Local charge
183         f = exp( - delta_c * (D_f * phi).^2) .^ D2s;
184         rho = c_s(p) * sinh(phi) .* f ./ (1 - nu + nu * f .* cosh(phi) );
185         exc = c_s(p) * cosh(phi) .* f ./ (1 - nu + nu * f .* cosh(phi) );
186
187
```

```
188         % Update residual
189         F1 = DF1_const * phi - 2.0 * delta_c * D2s * D_i * ...
190             (spdiags(exc, 0, N_z, N_z) * D_f * phi) - rho(1:end - 1);
191         F2 = F2_const + DF2_const * phi;
192         F = [F1; F2];
193         norm_res = norm(F, inf);
194         norm_delta_soln = norm(delta_soln, inf);
195         count = count + 1;
196     end
197
198     norm_res_store = norm_res;
199     norm_delta_soln_store = norm_delta_soln;
200
201     %reset loop
202     norm_res=1;
203     norm_delta_soln=1;
204     count=1;
205
206     alpha_count = alpha_count + 1;
207     if (alpha_count + 1 == alpha_max)
208         delta_c = alpha_store;
209     else
210         delta_c = delta_c + 0.01;
211     end
212
213     end
214
215     % ------------------------------------------------------------
216     % 3: Error checking
217     % ------------------------------------------------------------
218
219     % Throw a warning if the solution has not converged
220     if (norm_res_store > res_tol && norm_delta_soln_store > delta_norm_tol)
221             mesg_id = 'DDzPG:solutionNotConverged';
222             warning(mesg_id, 'NOT converged! res=%0.10f, delta_zeta=%0.10f', ...
223             norm_res_store, norm_delta_soln_store);
```

```
224     end
225
226 end
```

A.7 Auxillary programs for charge-induced thickening

A.7.1 Main program

The main program used for charge-induced thickening is identical to the one in Section A.6.1 (i.e. the main program in Section A.4.1 with the dielectric decrement modifications) except that it calls visChemiphoresis.m and visElectroosmosis.m instead of DDChemiphoresis.m and DDElectroosmosis.m respectively. In the case where Equation (6.5) is used instead of the general viscosity scale factor, the main program is identical to the one in Section A.4.1 except that it calls computeZetaPotentialBazant.m instead of computeZetaPotential.m. In addition, because we did not consider advection in the charge-induced thickening model, all references to p_i should be removed.

A.7.2 Modified zeta potential solver for Bazant's model

This program uses Equation (6.5). It is almost identical to computeZetaPotential.m, except that `res` and `J` should be replaced by the following (when ν is not 0)

```
1 res = Q - DeltaTheta * ( sign(V - Psi) .* sqrt( (2.0/nu) * c_s .* ...
2     log(1 + 2*nu*sinh(0.5*(V - Psi)).^2 ) ) .* sin(theta) );
3 J = - DeltaTheta * (...
4     sign(V - Psi) .* sqrt(nu * c_s/2.0) .* sinh(V - Psi) ...
5     .* sin(theta) ./ ...
6     ( ( 1 + 2.0 * nu * sinh(0.5*(V - Psi)).^2 ).* ...
7     sqrt( log(1 + 2*nu*sinh(0.5*(V - Psi)).^2 ) ) ) ...
8     );
```

A.7.3 Calculate viscosity scale factor: visScale.m

```
% visScale provides a scaling factor for the viscosity
%
% function [visSF] = visScale(c,c_infinity,nu)
%
% Inputs:
% c:            Local ionic concentration
% c_infinity:   Bulk ionic concentration
% nu:           Steric parameter
%
% Output:
% visSF:        Viscosity scaling factor

function [visSF] = visScale(c,c_infinity,nu)

% volume packing fraction
volFrac = c/2./c_infinity*nu*pi/6;
%volFrac = c/2./c_infinity*nu; %Use for testing with Eq (6.5)

% Uncomment the appropriate model

% (0) Constant viscosity: all 1's
visSF = volFrac./volFrac;

% (1) Model 1: Eq (4.32)
%visSF = (1./(1-volFrac./0.63)).^(2.5*0.63);

% (2) Model 2: Eq (4.33)
%F = volFrac + volFrac.^2 + volFrac.^3 - 2.3*volFrac.^4;
%visSF = (1+1.5*F)./(1-F);

% (3) Model 3: Bazant's model
%visSF = 1./(1-volFrac);
%change cosh to sinh in visElectroosmosis and visChemiphoresis
```

```
%because this model uses rho instead of c


%% Other models of interest

% Einstein
%visSF = 1 + 2.5*volFrac;

% Saito
%visSF = 1 + 2.5*volFrac./(1-volFrac);

% Gillespie
%visSF = (1 + volFrac/2)./(1-volFrac).^2;

end
```

A.7.4 Calculate diffusio-osmotic velocity: visChemiphoresis.m

```
% visChemiphoresis computes the diffusio-osmotic velocity at the surface of
% the particle in the charge-induced thickening case.
%
% function [u_cp, Du_cp_Dzeta] = visChemiphoresis(zeta, nu, res_tol)
%
% Inputs:
% zeta (required):      Zeta potential at the surface of the particle
% nu (required):        Steric parameter
% res_tol (optional):   Tolerance for residual in computation of w
%
% Outputs:
% u_cp:         Diffusio-osmotic velocity
% Du_cp_Dzeta:  Derivative of the diffusio-osmotic velocity with respect to
%   the zeta potential
% ------------------------------------------------------------------------

function [u_cp, Du_cp_Dzeta] = visChemiphoresis(zeta, nu, res_tol)
```

```
18
19 % ------------------------------------------------------------------
20 % STEP 1: check argument list and set default values
21 % ------------------------------------------------------------------
22
23 max_args = 3;
24 if (nargin < 2)
25     error('MATLAB:missingArgs','visChemiphoresis:missing arguments');
26 end
27 if (nargin < 3)
28     res_tol = 0.01;
29 end
30
31 % ------------------------------------------------------------------
32 % STEP 2: Initialization
33 % ------------------------------------------------------------------
34 num_gridpts = length(zeta);
35
36 % ------------------------------------------------------------------
37 % STEP 3: Sort the values of the zeta potential by ascending order
38 % ------------------------------------------------------------------
39
40 signZeta = sign(zeta);
41 [zeta_s, idx] = sort(abs(zeta));
42
43 % ------------------------------------------------------------------
44 % STEP 5: Perform the double integration
45 % ------------------------------------------------------------------
46
47 u_cp = spalloc(num_gridpts, 1, 0);
48 Du_cp_Dzeta = spalloc(num_gridpts, 1, 0);
49
50 v_min = eps;
51 res_w = 0.0;
52
53 for n = 1:num_gridpts
```

```
54
55      % Initialization
56      v_max = zeta_s(n);
57
58      % Evaluate the integral
59      if (v_min < 1E-5)
60          res_w = res_w + dblquad(@dbfun_v_small, 0, 1, v_min, 1E-5, res_tol) +...
61              dblquad(@dbfun_u_small, 0, 1E-5, 1E-5, v_max, res_tol) +...
62              dblquad(@dbfun, 1E-5, 1, 1E-5, v_max, res_tol);
63      else
64          res_w = res_w +...
65              dblquad(@dbfun_u_small, 0, 1E-5, v_min, v_max, res_tol) +...
66              dblquad(@dbfun, 1E-5, 1, v_min, v_max, res_tol);
67      end
68
69      % Derivative w.r.t the zeta potential
70      fun = @(u) dbfun(u, v_max);
71      fun_u_small = @(u) dbfun_u_small(u,v_max);
72      Du_cp_Dzeta(n) = quad(fun_u_small, 0, 1E-5, res_tol) +...
73          quad(fun, 1E-5, 1, res_tol);
74
75      % Update
76      u_cp(n) = res_w;
77      v_min = v_max;
78
79  end
80
81  % ------------------------------------------------------------
82  % STEP 6: Re-sort the arrays values
83  % ------------------------------------------------------------
84
85  u_cp(idx) = u_cp;
86  Du_cp_Dzeta(idx) = Du_cp_Dzeta;
87
88  % ------------------------------------------------------------
89  % Sub-functions: Define the functions to integrate
```

```
90 % ------------------------------------------------------------
91
92
93 function result = dbfun(u,v)
94     c_ratio = 2*cosh(v)./(1+2*nu*sinh(0.5 .* v).^2);
95     %change cosh to sinh for Model 3
96     visSF = visScale(c_ratio,1,nu);
97     if (nu == 0)
98         result = 0.5 .* v .* sinh(0.5 .* u .* v)./sinh(0.5 * v) ./ visSF;
99     else
100        result = 0.5 .* v .* sqrt( log(1 + 2 * nu * ...
101            sinh(0.5 .* u .* v).^2)./log(1 + 2 * nu * sinh(0.5 * v).^2)) ...
102            ./ visSF;
103    end
104 end
105
106 function result = dbfun_u_small(u,v)
107    c_ratio = 2*cosh(v)./(1+2*nu*sinh(0.5 .* v).^2);
108    %change cosh to sinh for Model 3
109    visSF = visScale(c_ratio,1,nu);
110    if (nu == 0)
111        result = 0.5 * v * sinh(0.5 .* u .* v)./sinh(0.5 .* v) ./ visSF;
112    else
113        result = 0.5 * v * sqrt(2*nu) .* sinh(0.5 .* u .* v) ./ ...
114            sqrt(log(1 + 2 * nu * sinh(0.5 * v).^2)) ./ visSF;
115    end
116 end
117
118 function result = dbfun_v_small(u,v)
119    c_ratio = 2*cosh(v)./(1+2*nu*sinh(0.5 .* v).^2);
120    %change cosh to sinh for Model 3
121    visSF = visScale(c_ratio,1,nu);
122    result = 0.5 * v .* sinh(0.5 .* u .* v) ./ sinh(0.5 .* v) ./ visSF;
123 end
124
125 end
```

A.7.5 Calculate electro-osmotic velocity: visElectroosmosis.m

```
% visElectroosmosis computes the electro-osmotic velocity at the surface of
% the particle in the charge-induced thickening case.
%
% function [u_el, Du_el_Dzeta] = visElectroosmosis(zeta, nu, res_tol)
%
% Inputs:
% zeta (required):      Zeta potential at the surface of the particle
% nu (required):        Steric parameter
% res_tol (optional):   Tolerance for residual in computation of w
%
% Outputs:
% u_el:         Electro-osmotic velocity
% Du_el_Dzeta:  Derivative of the electro-osmotic velocity with respect to
%   the zeta potential
% ------------------------------------------------------------------------------

function [u_el, Du_el_Dzeta] = visElectroosmosis(zeta, nu, res_tol)

% ------------------------------------------------------------
% STEP 1: check argument list and set default values
% ------------------------------------------------------------

max_args = 3;
if (nargin < 2)
    error('MATLAB:missingArgs','visElectroosmosis:missing arguments');
end
if (nargin < 3)
    res_tol = 0.01;
end

% ------------------------------------------------------------
% STEP 2: Initialization
% ------------------------------------------------------------
```

```
34 num_gridpts = length(zeta);
35
36 % ------------------------------------------------------------
37 % STEP 3: Sort the values of the zeta potential by ascending order
38 % ------------------------------------------------------------
39
40 signZeta = sign(zeta);
41 [zeta_s, idx] = sort(abs(zeta));
42
43
44 % ------------------------------------------------------------
45 % STEP 5: Perform the double integration
46 % ------------------------------------------------------------
47
48 u_el = spalloc(num_gridpts, 1, 0);
49 Du_el_Dzeta = spalloc(num_gridpts, 1, 0);
50
51 v_min = eps;
52 res_w = 0.0;
53
54 for n = 1:num_gridpts
55
56     % Initialization
57     v_max = zeta_s(n);
58
59     % Evaluate the integral
60     if (v_min < 5E-4)
61         res_w = res_w + dblquad(@dbfun_v_small, 0, 1, v_min, 5E-4, res_tol) +...
62             dblquad(@dbfun_u_small, 0, 5E-4, 5E-4, v_max, res_tol) +...
63             dblquad(@dbfun, 5E-4, 1, 5E-4, v_max, res_tol);
64     else
65         res_w = res_w +...
66             dblquad(@dbfun_u_small, 0, 5E-4, v_min, v_max, res_tol) +...
67             dblquad(@dbfun, 5E-4, 1, v_min, v_max, res_tol);
68     end
69
```

```
70      % Derivative w.r.t the zeta potential
71      fun = @(u) dbfun(u, v_max);
72      fun_u_small = @(u) dbfun_u_small(u,v_max);
73      Du_el_Dzeta(n) = quad(fun_u_small, 0, 5E-4, res_tol) +...
74          quad(fun, 5E-4, 1, res_tol);
75
76      % Update
77      u_el(n) = res_w;
78      v_min = v_max;
79
80  end
81
82  % ------------------------------------------------------------
83  % STEP 6: Re-sort the arrays values
84  % ------------------------------------------------------------
85
86  u_el(idx) = u_el;
87  u_el = u_el.*signZeta;
88  Du_el_Dzeta(idx) = Du_el_Dzeta;
89  Du_el_Dzeta = Du_el_Dzeta.*signZeta;
90  % Need to correct sign because function is even
91
92  % ------------------------------------------------------------
93  % Sub-functions: Define the functions to integrate
94  % ------------------------------------------------------------
95
96  function result = dbfun(u,v)
97      c_ratio = 2*cosh(v)./(1+2*nu*sinh(0.5 .* v).^2);
98      %change cosh to sinh for Model 3
99      visSF = visScale(c_ratio,1,nu);
100     if (nu == 0)
101         result = 0.25 * v .* sinh(u .* v)./sinh(0.5 * u .* v) ./ ...
102             sinh(0.5 * v) ./ visSF;
103     else
104         result = 0.5 * nu * v .* sinh(u .* v) ./ (1 + 2 * nu * ...
105             sinh(0.5 * u .* v).^2) ./ ...
```

```
106             sqrt( log(1 + 2 * nu * sinh(0.5 .* u .* v).^2) * ...
107             log (1 + 2 * nu * sinh(0.5 .* v).^2)) ./ visSF;
108     end
109 end
110
111 function result = dbfun_u_small(u,v)
112     c_ratio = 2*cosh(v)./(1+2*nu*sinh(0.5 .* v).^2);
113     %change cosh to sinh for Model 3
114     visSF = visScale(c_ratio,1,nu);
115     if (nu == 0)
116         result = 0.5 * v .* cosh(0.5 * u .* v) ./ sinh(0.5 * v) ./ visSF;
117     else
118         result = nu * v .* cosh(0.5 * u .* v) ./ (1 + 2 * nu * ...
119             sinh(0.5 * u .* v).^2) ./ sqrt(2*nu) ./ ...
120             sqrt (log (1 + 2 * nu * sinh(0.5 .* v).^2)) ./ visSF;
121     end
122 end
123
124 function result = dbfun_v_small(u,v)
125     c_ratio = 2;
126     visSF = visScale(c_ratio,1,nu);
127     holder = ones(1,length(u));
128     result = holder ./ visSF;
129 end
130
131 end
```

Bibliography

[1] M. M. Hatlo, R. H. H. G. Van Roij, and L. Lue. The electric double layer at high surface potentials: The influence of excess ion polarizability. *EPL (Europhysics Letters)*, 97(2):28010, 2012. doi:10.1209/0295-5075/97/28010.

[2] Kevin T Chu and Martin Z Bazant. Surface conservation laws at microscopically diffuse interfaces. *Journal of Colloid and Interface Science*, 315(1):319–329, 2007. doi:10.1016/j.jcis.2007.06.024.

[3] K. T. Chu and M. Z. Bazant. Nonlinear electrochemical relaxation around conductors. *Physical Review E*, 74:011501, 2006. doi:10.1103/PhysRevE.74.011501.

[4] M. S. Kilic, M. Z. Bazant, and A. Ajdari. Steric effects in the dynamics of electrolytes at large applied voltages. I. Double-layer charging. *Physical Review E*, 75:021502, 2007. doi:10.1103/PhysRevE.75.021502.

[5] M. S. Kilic and M. Z. Bazant. Steric effects in the dynamics of electrolytes at large applied voltages. II. Modified poisson-nernst-planck equations. *Physical Review E*, 75:021503, 2007. doi:10.1103/PhysRevE.75.021503.

[6] M. Z. Bazant, M. S. Kilic, B. D. Storey, and A. Ajdari. Towards and understanding of induced-charge electrokinetics at large applied voltages in concentrated solutions. *Advances in Colloid and Interface Science*, 152:48–88, 2009. doi:10.1016/j.cis.2009.10.001.

[7] M. Von Smoluchowski. Contribution à la théorie de l'endosmose électrique et de quelques phénomènes corrélatifs. *Bulletin de l'Académie des Sciences de Cracovie*, 8:182–200, 1903.

[8] F. A. Morrison Jr. Electrophoresis of a particle of arbitrary shape. *Journal of Colloid and Interface Science*, 34(2):210–214, 1970. doi:10.1016/0021-9797(70)90171-2.

[9] J. L. Anderson. Colloid transport by interfacial forces. *Annual Reviews of Fluid Mechanics*, 21:61–99, 1989. doi:10.1146/annurev.fluid.21.1.61.

[10] T. M. Squires and S. R. Quake. Microfluidics: Fluid physics at the nanoliter scale. *Reviews of Modern Physics*, 77:977–1026, July 2005. doi:10.1103/RevModPhys.77.977.

[11] H. A. Stone, A. D. Stroock, and A. Ajdari. Engineering flows in small devices: Microfluidics toward a lab-on-a-chip. *Annual Review of Fluid Mechanics*, 36: 381–411, 2004. doi:10.1146/annurev.fluid.36.050802.122124.

[12] J.-L. Viovy. Electrophoresis of DNA and other polyelectrolytes: Physical mechanisms. *Reviews of Modern Physics*, 72(3):813, 2000. doi:10.1103/RevModPhys.72.813.

[13] R. J. Hunter. *Foundations of Colloid Science*. Oxford University Press, 2001.

[14] J. Lyklema. *Fundamentals of Interface and Colloid Science: Solid-Liquid Interfaces*. Elsevier, 1995.

[15] J. J. Bikerman. Electrokinetic equations and surface conductance. a survey of the diffuse double layer theory of colloidal solutions. *Transactions of the Faraday Society*, 35:154–160, 1940. doi:10.1039/tf9403500154.

[16] J. Th. G. Overbeek. Theory of electrophoresis - the relaxation effect. *Kolloid-Beihefte*, 54:287–364, 1943.

[17] B. V. Derjaguin, S. S. Dukhin, and V. N. Shilov. Kinetic aspects of electrochemistry of disperse systems. part I. Introduction. *Advances in Colloid and Interface Science*, 13:141–152, 1980. doi:10.1016/0001-8686(80)87004-7.

[18] S. S. Dukhin and B. V. Derjaguin. Electrokinetic phenomena. In E. Matijevic and F. R. Eirich, editors, *Surface and Colloid Science*, volume 7. J. Wiley and Sons, 1974.

[19] S. S. Dukhin. Electrophoresis at large peclet numbers. *Advances in Colloid and Interface Science*, 36:219–248, 1991. doi:10.1016/0001-8686(91)80034-H.

[20] S. S. Dukhin and V. N. Shilov. Kinetic aspects of electrochemistry of disperse systems. part II. Induced dipole moment and the non-equilibrium double layer of a colloid particle. *Advances in Colloid and Interface Science*, 13:153–195, 1980. doi:10.1016/0001-8686(80)87005-9.

[21] S. S. Dukhin. Electrokinetic phenomena of the second kind and their applications. *Advances in Colloid and Interface Science*, 35:173–196, 1991. doi:10.1016/0001-8686(91)80022-C.

[22] S. Barany. Electrophoresis in strong electric fields. *Advances in Colloid and Interface Science*, 147-148:36–43, 2009. doi:10.1016/j.cis.2008.10.006.

[23] V. Shilov, S. Barany, C. Grosse, and O. Shramko. Field-induced disturbance of the double layer electro-neutrality and non-linear electrophoresis. *Advances in Colloid and Interface Science*, 104:159–173, 2003. doi:10.1016/S0001-8686(03)00040-X.

[24] N. A. Mishchuk and S. S. Dukhin. Electrophoresis of solid particles at large peclet numbers. *Electrophoresis*, 23:2012–2022, 2002. doi:10.1002/1522-2683(200207)23:13%3C2012::AID-ELPS2012%3E3.0.CO;2-Y.

[25] R. W. O'Brien and L. R. White. Electrophoretic mobility of a spherical colloidal particle. *J. Chem. Soc., Faraday Trans. 2*, 74:1607–1626, 1978. doi:10.1039/f29787401607.

[26] R. W. O'Brien and R. J. Hunter. The electrophoretic mobility of large colloidal particles. *Canadian Journal of Chemistry*, 59(13):1878–1887, 1981. doi:10.1139/v81-280.

[27] R. W. O'Brien and D. N. Ward. The electrophoresis of a spheroid with a thin double layer. *Journal of Colloid and Interface Science*, 121(2):402–413, 1988. doi:10.1016/0021-9797(88)90443-2.

[28] D.C. Prieve, J. L. Anderson, J. P. Ebel, and M. E. Lowell. Motion of a particle generated by chemical gradients. Part 2. Electrolytes. *Journal of Fluid Mechanics*, 148:247–269, 1984. doi:10.1017/S0022112084002330.

[29] B. Zaltzman and I. Rubinstein. Electro-osmotic slip and electroconvective instability. *Journal of Fluid Mechanics*, 579:173–226, 2007. doi:10.1017/S0022112007004880.

[30] I. Rubinstein and B. Zaltzman. Electro-osmotic slip of the second kind and instability in concentration polarization at electrodialysis membranes. *Mathematical Models and Methods in Applied Sciences*, 11(02):263–300, 2001. doi:10.1142/S0218202501000866.

[31] R. A. Rica and M. Z. Bazant. Electrodiffusiophoresis: Particle motion in electrolytes under direct current. *Physics of Fluids*, 22:112109, 2010. doi:10.1063/1.3496976.

[32] O. Schnitzer and E. Yariv. Macroscale description of electrokinetic flows at large zeta potentials: Nonlinear surface conduction. *Physical Review E*, 86(2): 021503, 2012. doi:10.1103/PhysRevE.86.021503.

[33] O. Schnitzer, R. Zeyde, I. Yavneh, and E. Yariv. Weakly nonlinear electrophoresis of a highly charged colloidal particle. *Physics of Fluids*, 25:052004, 2013. doi:10.1063/1.4804672.

[34] O. Schnitzer and E. Yariv. Strong-field electrophoresis. *Journal of Fluid Mechanics*, 701:333–351, 2012. doi:10.1017/jfm.2012.161.

[35] A. Ramos, H. Morgan, N. G. Green, and A. Castellanos. AC electric-field-induced fluid flow in microelectrodes. *Journal of Colloid and Interface Science*, 217(2):420–422, 1999. doi:10.1006/jcis.1999.6346.

[36] V. A. Murtsovkin. Nonlinear flows near polarized disperse particles. *Colloid Journal of the Russian Academy of Sciences*, 58(3):341–349, 1996.

[37] T. M. Squires and M. Z. Bazant. Induced-charge electro-osmosis. *Journal of Fluid Mechanics*, 509(1):217–252, 2004. doi:10.1017/S0022112004009309.

[38] M. Z. Bazant and T. M. Squires. Induced-charge electrokinetic phenomena: theory and microfluidic applications. *Physical Review Letters*, 92(6):066101, 2004. doi:10.1103/PhysRevLett.92.066101.

[39] E. Yariv. Nonlinear electrophoresis of ideally polarizable particles. *EPL*, 82: 54004, 2008. doi:10.1209/0295-5075/82/54004.

[40] J. J. Bikerman. Structure and capacity of electrical double layer. *Philosophical Magazine*, 33(220):384–397, 1942.

[41] O. Schnitzer and E. Yariv. Induced-charge electro-osmosis beyond weak fields. *Physical Review E*, 86(6):061506, 2012. doi:10.1103/PhysRevE.86.061506.

[42] M. Gouy. Sur la constitution de la charge électrique à la surface d'un électrolyte. *Journal de Physique Théorique et Appliquée*, 9(1):457–467, 1910. doi:10.1051/jphystap:019100090045700.

[43] D. L. Chapman. A contribution to the theory of electrocapillarity. *Philosophical Magazine*, 25(148):475–481, 1913. doi:10.1080/14786440408634187.

[44] O. Stern. The theory of the electrolytic double-layer. *Zeit. Elektrochem*, 30: 508–516, 1924.

[45] A. S. Khair and T. M. Squires. Ion steric effects on electrophoresis of a colloidal particle. *Journal of Fluid Mechanics*, 640:343, 2009. doi:10.1017/S0022112009991728.

[46] Franz Hofmeister. Zur lehre von der wirkung der salze. *Archiv fr experimentelle Pathologie und Pharmakologie*, 24(4-5):247–260, 1888. ISSN 0365-2009. doi:10.1007/BF01918191.

[47] D. Ben-Yaakov, D. Andelman, and R. Podgornik. Dielectric decrement as a source of ion-specific effects. *The Journal of Chemical Physics*, 134(7):074705–074705, 2011. doi:10.1063/1.3549915.

[48] John Barrett Hasted. *Aqueous dielectrics*, volume 17. Chapman and Hall London, 1973.

[49] E Glueckauf. Bulk dielectric constant of aqueous electrolyte solutions. *Transactions of the Faraday Society*, 60:1637–1645, 1964. doi:10.1039/tf9646001637.

[50] John J Molina, Sébastien Lectez, Sami Tazi, Mathieu Salanne, Jean-François Dufrêche, Jérôme Roques, Eric Simoni, Paul A Madden, and Pierre Turq. Ions in solutions: Determining their polarizabilities from first-principles. *The Journal of Chemical Physics*, 134:014511, 2011. doi:10.1063/1.3518101.

[51] John Jairo Molina. *Multi-scale modelling of ions in solution: from atomistic descriptions to chemical engineering*. PhD thesis, Université Paris VI Pierre et Marie Curie, 2013.

[52] Amir Levy, David Andelman, and Henri Orland. Dielectric constant of ionic solutions: A field-theory approach. *Physical Review Letters*, 108(22):227801, 2012. doi:10.1103/PhysRevLett.108.227801.

[53] V Gallardo, J Salcedo, P Vera, and AV Delgado. Electric and adsorption properties of pharmaceutical polymers. part i: Electrokinetics of aquacoat. *Colloid and Polymer Science*, 271(10):967–973, 1993. doi:10.1007/BF00654857.

[54] Teresa López-León, Ana B Jódar-Reyes, Delfi Bastos-González, and Juan L Ortega-Vinuesa. Hofmeister effects in the stability and electrophoretic mobility of polystyrene latex particles. *The Journal of Physical Chemistry B*, 107(24): 5696–5708, 2003. doi:10.1021/jp0216981.

[55] Andrea Salis, Francesca Cugia, Drew F Parsons, Barry W Ninham, and Maura Monduzzi. Hofmeister series reversal for lysozyme by change in ph and salt concentration: insights from electrophoretic mobility measurements. *Physical Chemistry Chemical Physics*, 14(13):4343–4346, 2012. doi:10.1039/c2cp40150a.

[56] D. Ben-Yaakov, D. Andelman, D. Harries, and R. Podgornik. Beyond standard poisson–boltzmann theory: ion-specific interactions in aqueous solutions. *Journal of Physics: Condensed Matter*, 21(42):424106, 2009. doi:10.1088/0953-8984/21/42/424106.

[57] Derek Frydel. Polarizable poisson–boltzmann equation: The study of polarizability effects on the structure of a double layer. *The Journal of Chemical Physics*, 134:234704, 2011. doi:10.1063/1.3598476.

[58] José Juan López-García, José Horno, and Constantino Grosse. Poisson–boltzmann description of the electrical double layer including ion size effects. *Langmuir*, 27(23):13970–13974, 2011. doi:10.1021/la2025445.

[59] H. Zhao and S. Zhai. The influence of dielectric decrement on electrokinetics. *Journal of Fluid Mechanics*, 724:69–94, 2013. doi:10.1017/jfm.2013.152.

[60] A. S. Khair and T. M. Squires. Fundamental aspects of concentration polarization arising from nonuniform electrokinetic transport. *Physics of Fluids*, 20(8): 087102–087102, 2008. doi:10.1063/1.2963507.

[61] M. S. Kilic. *Induced-charge Electrokinetics at Large Voltages*. PhD thesis, Masachusetts Institute of Technology, 2008.

[62] H. A. Stone and A. D. T. Samuel. Propulsion of microorganisms by surface distortions. *Physical Review Letters*, 77(19):4102–4104, 1996. doi:10.1103/PhysRevLett.77.4102.

[63] Brian J. Kirby. *Micro- and Nanoscale Fluid Mechanics: Transport in Microfluidic Devices*. Cambridge University Press, 2010.

[64] DA Haydon. A study of the relation between electrokinetic potential and surface charge density. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 258(1294):319–328, 1960. doi:10.1098/rspa.1960.0190.

[65] Aditya S Khair and John F Brady. Single particle motion in colloidal dispersions: a simple model for active and nonlinear microrheology. *Journal of Fluid Mechanics*, 557:73–118, 2006. doi:10.1017/S0022112006009608.

[66] Todd M Squires and Thomas G Mason. Fluid mechanics of microrheology. *Annual Review of Fluid Mechanics*, 42(1):413, 2009. doi:10.1146/annurev-fluid-121108-145608.

[67] J Lyklema. *Fundamentals of interface and colloid science: particulate colloids*. Academic Press, Waltham, 2005.

[68] Jonathan J Stickel and Robert L Powell. Fluid mechanics and rheology of dense suspensions. *Annual Review of Fluid Mechanics*, 37:129–149, 2005. doi:10.1146/annurev.fluid.36.050802.122132.

[69] Anna Marcinkowska-Gapińska, Jacek Gapinski, Waldemar Elikowski, Feliks Jaroszyk, and Leszek Kubisz. Comparison of three rheological models of shear flow behavior studied on blood samples from post-infarction patients. *Medical & Biological Engineering & Computing*, 45(9):837–844, 2007. doi:10.1007/s11517-007-0236-4.

[70] Heiko Stark and Stefan Schuster. Comparison of various approaches to calculating the optimal hematocrit in vertebrates. *Journal of Applied Physiology*, 113(3):355–367, 2012. doi:10.1152/japplphysiol.00369.2012.

[71] Irvin M Krieger and Thomas J Dougherty. A mechanism for non-newtonian flow in suspensions of rigid spheres. *Journal of Rheology*, 3:137, 1959. doi:10.1122/1.548848.

[72] Melvin Mooney. The viscosity of a concentrated suspension of spherical particles. *Journal of Colloid Science*, 6(2):162–170, 1951. doi:10.1016/0095-8522(51)90036-0.

[73] Eilers von H. Die viskosität von emulsionen hochviskoser stoffe als funktion der konzentration. *Kolloid-Zeitschrift*, 97(3):313–321, 1941. doi:10.1007/BF01503023.

[74] JS Chong, EB Christiansen, and AD Baer. Rheology of concentrated suspensions. *Journal of Applied Polymer Science*, 15(8):2007–2021, 1971. doi:10.1002/app.1971.070150818.

[75] Chingyi Chang and Robert L Powell. Hydrodynamic transport properties of concentrated suspensions. *AIChE Journal*, 48(11):2475–2480, 2002. doi:10.1002/aic.690481106.

[76] Andrew P Shapiro and Ronald F Probstein. Random packings of spheres and fluidity limits of monodisperse and bidisperse suspensions. *Physical Review Letters*, 68(9):1422, 1992. doi:10.1103/PhysRevLett.68.1422.

[77] RF Probstein, MZ Sengun, and TC Tseng. Bimodal model of concentrated suspension viscosity for distributed particle sizes. *Journal of Rheology*, 38:811, 1994. doi:10.1122/1.550594.

[78] Albert Einstein. A new determination of molecular dimensions. *Annalen der Physik*, 19(2):289–306, 1906.

[79] GK Batchelor. The stress system in a suspension of force-free particles. *Journal of Fluid Mechanics*, 41(3):545–570, 1970. doi:10.1017/S0022112070000745.

[80] GK Batchelor and JT Green. The hydrodynamic interaction of two small freely-moving spheres in a linear flow field. *Journal of Fluid Mechanics*, 56(2):375–400, 1972. doi:10.1017/S0022112072002927.

[81] GK Batchelor and JT Green. The determination of the bulk stress in a suspension of spherical particles to order c^2. *Journal of Fluid Mechanics*, 56(3): 401–427, 1972. doi:10.1017/S0022112072002435.

[82] GK Batchelor. The effect of brownian motion on the bulk stress in a suspension of spherical particles. *Journal of Fluid Mechanics*, 83(1):97–117, 1977. doi:10.1017/S0022112077001062.

[83] B Cichocki, ML Ekiel-Jeżewska, and E Wajnryb. Three-particle contribution to effective viscosity of hard-sphere suspensions. *The Journal of Chemical Physics*, 119:606, 2003. doi:10.1063/1.1484380.

[84] Eugene Guth and R Simha. Untersuchungen über die viskosität von suspensionen und lösungen. 3. über die viskosität von kugelsuspensionen. *Colloid & Polymer Science*, 74(3):266–275, 1936. doi:10.1007/BF01428643.

[85] Vladimir Vand. Viscosity of solutions and suspensions. i. theory. *The Journal of Physical Chemistry*, 52(2):277–299, 1948. doi:10.1021/j150458a001.

[86] Robert Simha. A treatment of the viscosity of concentrated suspensions. *Journal of Applied Physics*, 23(9):1020–1024, 1952. doi:10.1063/1.1702338.

[87] GJ Kynch. The effective viscosity of suspensions. *British Journal of Applied Physics*, 5(S3):S5–S10, 1954. doi:10.1088/0508-3443/5/S3/303.

[88] GJ Kynch. The effective viscosity of suspensions of spherical particles. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 237(1208):90–116, 1956. doi:10.1098/rspa.1956.0164.

[89] William B Russel. The rheology of suspensions of charged rigid spheres. *Journal of Fluid Mechanics*, 85(2):209–232, 1978. doi:10.1017/S0022112078000609.

[90] William B Russel. The huggins coefficient as a means for characterizing suspended particles. *Journal of the Chemical Society, Faraday Transactions 2*, 80 (1):31–41, 1984. doi:10.1039/f29848000031.

[91] WB Russel and AP Gast. Nonequilibrium statistical mechanics of concentrated colloidal dispersions: Hard spheres in weak flows. *The Journal of Chemical Physics*, 84:1815, 1986. doi:10.1063/1.450428.

[92] Cristina Urdaneta Thomas and M Muthukumar. Three-body hydrodynamic effects on viscosity of suspensions of spheres. *The Journal of Chemical Physics*, 94:5180, 1991. doi:10.1063/1.460555.

[93] John F Brady. The rheological behavior of concentrated colloidal dispersions. *Journal of Chemical Physics*, 99(1):567–581, 1993. doi:10.1063/1.465782.

[94] CWJ Beenakker. The effective viscosity of a concentrated suspension of spheres (and its relation to diffusion). *Physica A: Statistical Mechanics and its Applications*, 128(1):48–81, 1984. doi:10.1016/0378-4371(84)90081-5.

[95] F Booth. The electroviscous effect for suspensions of solid spherical particles. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 203(1075):533–551, 1950. doi:10.1098/rspa.1950.0155.

[96] DA Saville. Electrokinetic effects with small particles. *Annual Review of Fluid Mechanics*, 9(1):321–337, 1977. doi:10.1146/annurev.fl.09.010177.001541.

[97] William B Russel. Bulk stresses due to deformation of the electrical double layer around a charged sphere. *Journal of Fluid Mechanics*, 85(04):673–683, 1978. doi:10.1017/S0022112078000853.

[98] Grinnell Jones and Malcolm Dole. The viscosity of aqueous solutions of strong electrolytes with special reference to barium chloride. *Journal of the American Chemical Society*, 51(10):2950–2964, 1929. doi:10.1021/ja01385a012.

[99] Manfred Kaminsky. Experimentelle untersuchungen über die konzentrations- und temperaturabhängigkeit der zähigkeit wäßriger lösungen starker elektrolyte. *Zeitschrift für Physikalische Chemie*, 12(3_4):206–231, 1957. doi:10.1524/zpch.1957.12.3_4.206.

[100] MCS Subha, K Chowdoji Rao, and S Brahmaji Rao. Relative viscosities of metal formates in formic acid. *Journal of the Indian Institute of Science*, 66: 397–403, 1986.

[101] LC Heda, Rashmi Sharma, and Pramod B Chaudhari. Viscometric investigations of some derivatives of 5-substituted indole dihydropyrimidines 2 - ones in mixed organic solvents. *International Journal of Chemical Sciences*, 7(3): 1595–1605, 2009. doi:10.1590/S0100-46702010000200003.

[102] James F Skinner and Raymond M Fuoss. Electrostriction in polar solvents. ii. *The Journal of Physical Chemistry*, 68(10):2998–3003, 1964. doi:10.1021/j100792a045.

[103] JE Desnoyers and G Perron. The viscosity of aqueous solutions of alkali and tetraalkylammonium halides at 25°c. *Journal of Solution Chemistry*, 1(3):199–212, 1972. doi:10.1007/BF00645101.

[104] Antonio Sacco, Giuseppe Petrella, Mario Della Monica, and Maurizio Castagnolo. Electrolyte viscosities in hexamethylphosphotriamide at 25$^{c}ircc$. *Journal of the Chemical Society, Faraday Transactions 1: Physical Chemistry in Condensed Phases*, 73:1936–1942, 1977. doi:10.1039/f19777301936.

[105] Giuseppe Petrella and Antonio Sacco. Viscosity and conductance studies in ethylene carbonate at 40°c. *Journal of the Chemical Society, Faraday Transactions 1: Physical Chemistry in Condensed Phases*, 74:2070–2076, 1978. doi:10.1039/f19787402070.

[106] Peter T Thompson, Barbara Fisher, and Robert H Wood. Viscosities of solutions of electrolytes and non-electrolytes in ethylene carbonate at 40°c. *Journal of Solution Chemistry*, 11(1):1–15, 1982. doi:10.1007/BF00664330.

[107] Idar Svorstøl, Tove Sigvartsen, and Jon Songstad. Solvent properties of dichloromethane. vii. viscosity studies of electrolytes in dichloromethane. *Acta Chemica Scandinavia B42*, 42b:133–144, 1988. doi:10.3891/acta.chem.scand.42b-0133.

[108] DJP Out and JM Los. Viscosity of aqueous solutions of univalent electrolytes from 5 to 95°c. *Journal of Solution Chemistry*, 9(1):19–35, 1980. doi:10.1007/BF00650134.

[109] Jaka Horvat, Marija Bešter-Rogač, Cveto Klofutar, and Darja Rudan-Tasic. Viscosity of aqueous solutions of lithium, sodium, potassium, rubidium and caesium cyclohexylsulfamates from 293.15 to 323.15 k. *Journal of Solution Chemistry*, 37(9):1329–1342, 2008. doi:10.1007/s10953-008-9311-1.

[110] Yizhak Marcus. Effect of ions on the structure of water: structure making and breaking. *Chemical Reviews*, 109(3):1346–1370, 2009. doi:10.1021/cr8003828.

[111] H Donald B Jenkins and Yizhak Marcus. Viscosity b-coefficients of ions in solution. *Chemical Reviews*, 95(8):2695–2724, 1995. doi:10.1021/cr00040a004.

[112] Amalendu Chandra and Biman Bagchi. Ionic contribution to the viscosity of dilute electrolyte solutions: Towards a microscopic theory. *The Journal of Chemical Physics*, 113:3226, 2000. doi:10.1063/1.1286963.

[113] Claudio Contreras-Aburto and Gerhard Nägele. Viscosity of electrolyte solutions: a mode-coupling theory. *Journal of Physics: Condensed Matter*, 24(46): 464108, 2012. doi:10.1088/0953-8984/24/46/464108.

[114] K Ibuki and M Nakahara. Dielectric friction theory of the viscosity of electrolyte solutions. *The Journal of Chemical Physics*, 85:7312, 1986. doi:10.1063/1.451370.

[115] Barbara Hribar, Noel T Southall, Vojko Vlachy, and Ken A Dill. How ions affect the structure of water. *Journal of the American Chemical Society*, 124 (41):12302–12311, 2002. doi:10.1021/ja026014h.

[116] Nobuhiko Saitô. Concentration dependence of the viscosity of high polymer solutions. *Journal of the Physical Society of Japan*, 5(1):4–8, 1950. doi:10.1143/JPSJ.5.4.

[117] Nobuhiko Saitô. A remark on the hydrodynamical theory of the viscosity of solutions of macromolecules. *Journal of the Physical Society of Japan*, 7(5): 447–450, 1952. doi:10.1143/JPSJ.7.447.

[118] D Bedeaux. The effective viscosity for a suspension of spheres. *Journal of Colloid and Interface Science*, 118(1):80–86, 1987. doi:10.1016/0021-9797(87)90436-X.

[119] JC Van der Werff, CG De Kruif, C Blom, and J Mellema. Linear viscoelastic behavior of dense hard-sphere dispersions. *Physical Review A*, 39(2):795, 1989. doi:10.1103/PhysRevA.39.795.

[120] Anthony JC Ladd. Hydrodynamic transport coefficients of random dispersions of hard spheres. *The Journal of Chemical Physics*, 93:3484, 1990. doi:10.1063/1.458830.

[121] GK Batchelor. Sedimentation in a dilute polydisperse system of interacting spheres. part 1. general theory. *Journal of Fluid Mechanics*, 119(1):379–408, 1982. doi:10.1017/S0022112082001402.

[122] J Bergenholtz, JF Brady, and M Vicic. The non-newtonian rheology of dilute colloidal suspensions. *Journal of Fluid Mechanics*, 456:239–275, 2002. doi:10.1017/S0022112001007583.

[123] Li Yuan-Hui and Sandra Gregory. Diffusion of ions in sea water and in deep-sea sediments. *Geochimica et Cosmochimica Acta*, 38(5):703–714, 1974. doi:10.1016/0016-7037(74)90145-8.

[124] S Koneshan, Jayendran C Rasaiah, RM Lynden-Bell, and SH Lee. Solvent structure, dynamics, and ion mobility in aqueous solutions at 25 c. *The Journal of Physical Chemistry B*, 102(21):4193–4204, 1998. doi:10.1021/jp980642x.

[125] M. Flury and T. Gimmi. Solute diffusion. In J. H. Dane and G. C. Topp, editors, *Methods of Soil Analysis Part 4*, pages 1323–1551. Soil Science Society of America, Madison, WI, 2002.

[126] Robert J. L. Chimenti. Electrophoretic separation, April 21 1992. US Patent 5,106,468.

[127] A. S. Dukhin and Dukhin S. S. Aperiodic capillary electrophoresis method using an alternating current electric field for separation of macromolecules. *Electrophoresis*, 26:2149–2153, 2005. doi:10.1002/elps.200410408.

[128] B. Neirinck, J. Fransaer, O. Van der Biest, and . Vleugels. Aqueous electrophoretic deposition in asymmetric ac electric fields. *Electrochemistry Communications*, 11:57–60, 2009. doi:10.1016/j.elecom.2008.10.028.

[129] M. Z. Bazant, B. D. Storey, and A. A. Kornyshev. Double layer in ionic liquids: Overscreening versus crowding. *Physical Review Letters*, 106(4):046102, 2011. doi:10.1103/PhysRevLett.106.046102.

[130] B. D. Storey and M. Z. Bazant. Effects of electrostatic correlations on electrokinetic phenomena. *Physical Review E*, 86(5):056303, 2012. doi:10.1103/PhysRevE.86.056303.