Computational Electrokinetics

Roman Zeyde

Computational Electrokinetics

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science

Roman Zeyde

Submitted to the Senate of
the Technion — Israel Institute of Technology
Adar 5773 Haifa March 2013

The research thesis was done under the supervision of Prof. Irad Yavneh in the Computer Science Department.

Acknowledgment

I would like to thank my adviser, Prof. Irad Yavneh for his outstanding guidance, helpful discussions and valuable advice. I am deeply grateful to my family, and especially to my wife, Galit, for their love and support.

The generous financial support of the Technion is gratefully acknowledged.

Contents

List of Figures

Abstract

In this work we study the electrokinetic migration of particles in an electrolyte solution due to the application of an external electric field, and propose a numerical framework for the iterative solution of such problems. The electrokinetic transport process can be used for transporting and manipulating micro- and nanoscale objects in many nanotechnology applications, nano-fluidic devices, packed-bed separation, desalination processes and various electrophoresis applications. Due to strong electrostatic forces, a thin boundary layer forms near the particle surface. This results in scale disparity of the boundary layer, which makes a full numerical solution challenging. We employ nonlinear macroscale effective boundary conditions that have been derived using the specific chemical properties of the particle (for an ion-exchanger particle by Yariv in [1] and for a surface charged inert particle by Schnitzer and Yariv in [2]). The resulting macroscale nonlinear partial differential system is discretized and an iterative Newton solver is constructed automatically from the discrete equations. Numerical results are obtained for an ion-exchanger and for a surface charged inert particle. An asymptotic analytical solution is used for the validation of the solver. The numerical results are compared to the asymptotic solutions, and good correspondence is achieved. Finally, the new solver is applied in the regime of moderately large values of the external field, where no analytical results are known. The numerical results uncover the strong-field scaling behavior, including a new boundary layer at the front of the particle that scales exponentially with respect to the external field, and also a well-defined dependence of the particle steady-state velocity on this field.

Abbreviations and Notations

$\varPhi$	—	Electric potential
C	—	Ionic concentration
$\boldsymbol{V}$	—	Fluid velocity
P	—	Fluid pressure
T	—	Stress tensor
$\boldsymbol{F}$	—	Force vector
$\hat{\boldsymbol{n}}$	—	Surface normal
$\mathcal{O}$	—	Operator
β	—	Dimensionless electric field magnitude
δ	—	Dimensionless Debye layer width
$\mathscr{U}$	—	Dimensionless steady-state velocity

Chapter 1

Introduction

1.1 The Physical Problem

1.1.1 Electrokinetic Phenomena

Electrokinetic theory describes the dynamics of charged particles in ionic fluids [3, 4]. When a particle acquires surface charge, a layer of ions of opposite charge is attracted to the surface via electric forces, creating a double-layer structure around the particle (see Figure 1.1). This structure, called “Debye layer”, electrically screens the surface charge, creating a potential difference between the particle and the outer layer of the fluid bulk. In cases where the layer width is much smaller than the particle size, an analytical asymptotic solution for the Debye layer dynamics can be derived [5].

The variables of the electrokinetic flow equations are the electrostatic potential φ, fluid velocity $\boldsymbol{v}$ and its pressure p, and ionic concentration c. The boundary conditions are determined by the specific problem under consideration and defined by the particle’s geometry, chemical characteristics, and the fluid dynamics. The partial differential equations that describe the system dynamics under an external electric field, are coupled and nonlinear, and in general have no analytic solution. Moreover, any numerical solver must handle the scale disparity caused by the Debye layer width being much smaller than the particle size. An effective boundary condition outside the Debye layer has been developed for spherical particles, yielding a macroscale model in [1, 2]. The model is solved for weak electric field in an axisymmetric setting but it is hard to extend this analytic solution to more general systems. Once the electric field becomes stronger, significant nonlinear phenomena emerge. This interesting regime has not yet been explored extensively.

1.1.2 Applications

Recently, it has been conceived that such type of phenomena can be very useful in microfluidics devices and methods for controlling and manipulating fluid flows in micro- and nanoscale. Many different components are used in microfluidics such as valves, pumps, sensors, mixers, filters, separators, heaters etc., and are combined into lab-on-a-chip systems.

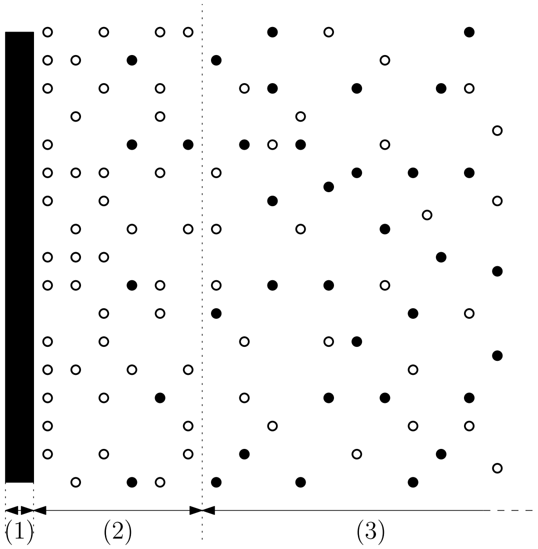

Figure 1.1: Schematic structure of the double layer: (1) charged particle surface, (2) Debye layer, (3) fluid bulk. If the particle surface has positive (black) surface charge, it attracts negative (white) ions from the fluid making the Debye layer negatively charged (as opposed to the rest of the fluid bulk, which is electrically neutral). The zeta potential is defined as the voltage drop across the Debye layer (2).

The manipulation can be achieved by either applying the external field at inlets and outlets, or it can be applied locally in the microchannel by integrated components. A typical microchannel is produced with lithographic techniques. Normally the Reynolds number is very low (on the order of unity or smaller) which means that viscosity dominates and the convective term of the Navier-Stokes equations is insignificant. This also means that microflows most often are laminar.

In the last decade there has been an increase in microfluidics research, due to the spread of tools for microfluidic systems fabrication, production of cheap and portable devices for fast medical analysis, and fundamental research of physical, chemical and biological processes. Various applications are [6, 7, 8, 9]:

1. DNA separation and sequencing [10].

2. Cell analysis and forensic identification [11, 12, 13].

3. Integrated microchemical systems [14].

4. Packed-bed separation [15, 16].

5. Desalination processes [17].

6. Drug delivery and pathogen detection [18].

Because the nonlinear macroscale model for these electrokinetic phenomena has no closed-form analytic solution, a numerical solver for this system is of importance.

1.2 Governing Equations

1.2.1 Nondimensionalization

We follow the derivation presented by Yariv in [5]. Spatial coordinates $\boldsymbol{r}$ are normalized by a^* (the characteristic size of the particle). Positive and negative ionic concentrations are denoted by c_+ and c_-, respectively, and are normalized by the ambient ionic concentration c^*. The ionic valences are $\pm z$. The dimensional ionic fluxes, $\boldsymbol{j}_\pm$, are given by the Nernst-Plank equation (using the Einstein relation for the electric mobility, $\mu^*_q = D^*/k^*_B T^*$, where D^* is the ionic diffusivity):

$$\boldsymbol{j}^*_\pm = -D^*\boldsymbol{\nabla} c^*_\pm + \boldsymbol{v}^* c^*_\pm \mp \frac{ze^* D^*}{k^*_B T^*} c^*_\pm \boldsymbol{\nabla}\varphi^*, \tag{1.1}$$

The electric potential φ is normalized by the thermal voltage: $\varphi^* = k^*_B T^*/ze^* = R^* T^*/zF^*$, where $R^* = k^*_B N_A$, $F^* = e^* N_A$ is the Faraday number and N_A is the Avogadro constant.

The dimensional Stokes equations with electrostatic forces are given by:

$$\boldsymbol{\nabla}\cdot\boldsymbol{v} = 0, \quad -\boldsymbol{\nabla} p^* + \mu^*\boldsymbol{\nabla}^2\boldsymbol{v}^* + \epsilon^*\nabla^2\varphi^*\boldsymbol{\nabla}\varphi^* = 0. \tag{1.2}$$

The velocity $\boldsymbol{v}$ is normalized by $v^* = \epsilon^*(\varphi^*)^2/a^*\mu^*$, and the pressure p is normalized by $p^* = \mu^* v^*/a^* = \epsilon^*(\varphi^*)^2/(a^*)^2$. The fluxes $\boldsymbol{j}^*_\pm$ are normalized by $j^* = D^* c^*/a^*$.

The dimensional Poisson equation can be written as

$$(c^*_+ - c^*_-)ze^* = -\epsilon^*\nabla^2\varphi^*, \tag{1.3}$$

After nondimensionalization, the Nernst-Plank, Stokes and Poisson equations read:

$$\boldsymbol{j}_\pm = -\boldsymbol{\nabla} c_\pm + \alpha\boldsymbol{v} c_\pm \mp c_\pm\boldsymbol{\nabla}\varphi, \tag{1.4}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{v} = 0, \quad -\boldsymbol{\nabla} p + \boldsymbol{\nabla}^2\boldsymbol{v} + \nabla^2\varphi\boldsymbol{\nabla}\varphi = 0, \tag{1.5}$$

$$c_+ - c_- = -2\delta^2\nabla^2\varphi, \tag{1.6}$$

where $\delta = \delta^*/a^*$ is the dimensionless Debye thickness ($\delta^* = \sqrt{\epsilon^*\varphi^*/2c^*e^*z}$) and $\alpha = v^*a^*/D^* = \epsilon^*(\varphi^*)^2/\mu^*D^*$ is the dimensionless Peclet number. The force on the particle is denoted by $\boldsymbol{F}$ normalized by $\mu^*v^*a^* = \epsilon^*(\varphi^*)^2$. The stress tensor is denoted by T and normalized by $p^* = \mu^*v^*/a^*$. The electric field is denoted by $\boldsymbol{E}$ and normalized by $E^* = \varphi^*/a^*$. The surface charge is denoted by σ and normalized by $\sigma^* = \epsilon^*\varphi^*/\delta^*$.

The ionic fluxes are given by ion diffusion, electrostatic forces and ion advection by the fluid in (1.4). Due to conservation of ion concentrations, the fluxes are divergence-free:

$$\boldsymbol{\nabla}\cdot\boldsymbol{j}_\pm = 0.$$

These equations may be re-written by using salt $c = (c_+ + c_-)/2$ and charge $q = (c_+ - c_-)/2$. Thus, salt and charge fluxes are defined by:

$$\boldsymbol{j} = \tfrac{\boldsymbol{j}_+ + \boldsymbol{j}_-}{2} = -\boldsymbol{\nabla} c - q\boldsymbol{\nabla}\varphi + \alpha \boldsymbol{v} c, \quad \boldsymbol{i} = \tfrac{\boldsymbol{j}_+ - \boldsymbol{j}_-}{2} = -\boldsymbol{\nabla} q - c\boldsymbol{\nabla}\varphi + \alpha \boldsymbol{v} q, \tag{1.7}$$

and are divergence-free as well:

$$\boldsymbol{\nabla}\cdot\boldsymbol{j} = 0, \quad \boldsymbol{\nabla}\cdot\boldsymbol{i} = 0. \tag{1.8}$$

In addition, the charge concentration is governed by the Poisson equation (1.6)

$$q = -\delta^2\nabla^2\varphi. \tag{1.9}$$

1.2.2 Boundary conditions

We assume that a constant electric field of nondimensional magnitude β, is applied to the system, in the direction denoted by the unit vector $\hat{\boldsymbol{\imath}}$. As a result, the particle will equilibrate at a non-zero steady-state velocity $\mathscr{U}\hat{\boldsymbol{\imath}}$, which is a function of the applied field, and is not known a priori.

We employ the reference frame of the particle. Far away from the particle, we have a uniform applied electrostatic field, uniform flow, and ambient ionic concentrations:

$$\boldsymbol{v} \to -\mathscr{U}\hat{\boldsymbol{\imath}}, \quad \boldsymbol{\nabla}\varphi \to -\beta\hat{\boldsymbol{\imath}}, \quad c_\pm \to 1. \tag{1.10}$$

We consider a spherical particle. The chemical properties of the particle's surface $\mathcal{S}$ (defined at $r = 1$ with normal $\hat{\boldsymbol{n}} = \hat{\boldsymbol{r}}$), determine the boundary conditions on $\mathcal{S}$. We consider the following scenarios:

Ion-exchanger

We follow the microscale model presented at [1] and assume high conductance of the particle (yielding a uniform electric potential), zero slip, anion impermeability, cation selectivity (with fast Butler-Volmer[19] kinetics $\hat{\boldsymbol{n}}\cdot\boldsymbol{j}_+ = k(1 - c_+/\gamma)$, where $k \to \infty$):

$$\varphi = \mathscr{V}, \quad \boldsymbol{v} = \boldsymbol{0}, \quad \hat{\boldsymbol{n}}\cdot\boldsymbol{j}_- = 0, \quad c_+ = \gamma, \tag{1.11}$$

where γ is the ratio between the cationic concentration on the surface and the cationic concentration in the fluid bulk and $\mathscr{V}$ is an unknown constant electric potential at the ion-exchanger surface.

Inert particle

We follow the microscale model presented at [2] and assume constant surface charge zero slip and ion impermeability:

$$\hat{\boldsymbol{n}}\cdot(\boldsymbol{\nabla}\varphi - \chi\boldsymbol{\nabla}\varphi_s) = \delta^{-1}\sigma, \quad \boldsymbol{v} = \boldsymbol{0}, \quad \hat{\boldsymbol{n}}\cdot\boldsymbol{j}_\pm = 0, \tag{1.12}$$

where χ is the ratio between the dielectric constants of the particle and the fluid.

1.2.3 Separation of scales

When $\delta \ll 1$, the fluid remains electroneutral ($q \approx 0$, thus $c_+ \approx c_-$), except inside a thin boundary layer of width $O(\delta)$ surrounding the particle (the "Debye layer"). This boundary layer makes it very hard to construct a direct numerical solver for the full electrokinetic problem. However, the scale disparity at the Debye layer can be exploited (as shown by Yariv in [5]), which allows the PDE system to be rewritten in the "inner" region as an ODE system using the asymptotic limit of $\delta \to 0$. This system can be integrated and solved analytically, yielding "effective" boundary conditions near the particle surface for the "outer" bulk region equations.

Bulk-scale equations

The Nernst-Planck equations above (1.7) for the fluid bulk outside the boundary layer [5], using bulk variables denoted by uppercase letters, read:

$$Q = 0, \quad C = C_+ = C_-, \quad \boldsymbol{J} = -\boldsymbol{\nabla} C + \alpha \boldsymbol{V} C, \quad \boldsymbol{I} = -C \boldsymbol{\nabla} \varPhi. \tag{1.13}$$

Because the flow is incompressible ($\boldsymbol{\nabla} \cdot V = 0$), ion conservation results in the following salt and charge conservation equations – the salt and charge fluxes above (1.13) are divergence-free (1.8):

$$\nabla^2 C - \alpha \boldsymbol{V} \cdot \boldsymbol{\nabla} C = 0, \quad \boldsymbol{\nabla} \cdot (C \boldsymbol{\nabla} \varPhi) = 0. \tag{1.14}$$

The macroscale Stokes flow equations have the same form as before (1.5):

$$\boldsymbol{\nabla} \cdot \boldsymbol{V} = 0, \quad \nabla^2 \boldsymbol{V} - \boldsymbol{\nabla} P + \boldsymbol{\nabla} \varPhi \nabla^2 \varPhi = \boldsymbol{0}. \tag{1.15}$$

1.2.4 Bulk-scale effective boundary conditions

By integrating the equations across the Debye layer, the effective boundary conditions at the particle surface are:

Ion-exchanger

See [1] for full derivation of the nonlinear effective boundary conditions (and also Appendix A):

$$\begin{aligned} 0 &= \varPhi + \log C, \\ 0 &= \frac{\partial}{\partial n} (\varPhi - \log C), \\ \boldsymbol{V} &= \zeta \cdot \boldsymbol{\nabla}_{\mathcal{S}} \varPhi + 2 \log \left(1 - \tanh^2 \frac{\zeta}{4}\right) \cdot \boldsymbol{\nabla}_{\mathcal{S}} \log C, \end{aligned} \tag{1.16}$$

where $\zeta = \mathscr{V} - \varPhi = \log(C/\gamma)$ is the zeta potential.

Highly charged inert particle

See [2] for full derivation of the nonlinear effective boundary conditions. Note that due to surface conduction, the resulting boundary conditions are:

$$\begin{aligned}
0 &= \frac{\partial C}{\partial n} - \mathrm{Du}\nabla_s^2\left(\varPhi - \log C\right),\\
0 &= \frac{\partial C}{\partial n} + C\frac{\partial \varPhi}{\partial n},\\
\boldsymbol{V} &= \zeta \cdot \boldsymbol{\nabla}_{\mathcal{S}}\left(\varPhi - \log C\right) + 4\log 2 \cdot \boldsymbol{\nabla}_{\mathcal{S}} \log C,
\end{aligned} \tag{1.17}$$

where $\zeta = \bar{\zeta} - \log C$ is the zeta potential and $\bar{\zeta} = 2\log\sigma$.

1.2.5 Steady-state velocity

The stress tensor in the fluid is composed of Newtonian and Maxwel stresses:

$$\mathsf{T} = \boldsymbol{\nabla}\boldsymbol{V} + (\boldsymbol{\nabla}\boldsymbol{V})^{\dagger} - P\mathsf{I} + \boldsymbol{\nabla}\varPhi\boldsymbol{\nabla}\varPhi - \frac{1}{2}(\boldsymbol{\nabla}\varPhi \cdot \boldsymbol{\nabla}\varPhi). \tag{1.18}$$

In steady state (constant particle drift velocity $\mathscr{U}$), the total force acting on the particle (computed by the following surface integral) must vanish:

$$\boldsymbol{F}(\beta, \mathscr{U}) = \oint_{\mathcal{S}} \mathsf{T} \cdot \hat{\boldsymbol{n}} dA = \boldsymbol{0}. \tag{1.19}$$

The total force $\boldsymbol{F}(\beta, \mathscr{U})$ is a function of the applied electric field β and the velocity $\mathscr{U}$. For any given electric field β, the force-free constraint above can be solved numerically for $\mathscr{U}$ to yield the corresponding steady-state velocity.

1.3 Summary of achievements in this work

In this work, we develop and implement a novel software framework, that enables the generation of iterative numerical solvers for nonlinear macroscale electrokinetic problems. A novel feature is the combination of a numerical framework with an analytical solution of the thin Debye layer, that is available from the literature and provides effective boundary conditions. Therefore, the Debye layer does not need to be resolved by the computational grid, saving a tremendous amount of numerical resources.

We apply this framework to the study of systems that have no closed-form solutions, such as ion-exchanger migration [1] and the electrophoresis problem [2] for a spherical particle in an infinite domain. The numerical results can be used to gain insight into the chemical and physical behavior in far more general regimes than are currently well-understood, and also to possibly lead to further analytical developments, based on new scaling behavior that is discovered numerically. We have also discovered interesting physical phenomena that may be observed experimentally. One such phenomenon is nonlinear electrokinetic vortex flow, which is useful for microfluidic mixing applications [20, 21].

In addition, we develop and present an asymptotic derivation for the ion-exchanger migration problem up to third order terms. This new analytical result is also used to verify the numerical framework and the numerical results.

Chapter 2

Numerical Scheme

2.1 Numerical Framework

We start by presenting the numerical framework that is used for the construction of the nonlinear solver.

2.1.1 Finite Volume Method

Finite Volume methods [22] are used to define a discretization of the conservation equations' integral forms. The simulation domain is divided into a set of discrete non-overlapping control volumes. Each conservation equation is discretized over the control volumes, such that the integral form of the conservation law is satisfied using an approximation of the fluxes over the control volume faces. This method ensures that the discretization is conservative in the discrete sense.

2.1.2 Flux Discretization

In order to define the fluxes on the boundaries of each control volume, we employ the conservative formulation of the governing equations above. The flux terms are defined using finite differences and linear interpolation of the discrete variables (defined at the centroid of each cell), except for the advection flux which should not be discretized using linear interpolation – due to a numerical instability for large cell Peclet numbers [23].

2.1.3 Ghost Points

In order to discretize boundary condtions for Φ, C and $\boldsymbol{V}$, "ghost" points are employed. The grid is extended to include points outside the domain interior, and the variables at the "ghost" points are set to satisfy the discretized boundary conditions on the grid boundaries. Note that P has no boundary conditions – so the pressure variable is defined only in the domain interior.

2.1.4 Newton's Method

Newton's method is used to solve a nonlinear system of equations of the form $\mathcal{O}(\boldsymbol{x}) = \mathbf{0}$. For small step $\Delta\boldsymbol{x}$, we can linearize the operator by

$$\mathcal{O}(\boldsymbol{x} + \Delta\boldsymbol{x}) \approx \mathcal{O}(\boldsymbol{x}) + \boldsymbol{\nabla}\mathcal{O}(\boldsymbol{x})\Delta\boldsymbol{x} \tag{2.1}$$

Thus, we use the following iteration, starting from an initial solution $\boldsymbol{x}_0$:

$$\Delta\boldsymbol{x}_n = -\left[\boldsymbol{\nabla}\mathcal{O}(\boldsymbol{x}_n)\right]^{-1} \cdot \mathcal{O}(\boldsymbol{x}_n) \tag{2.2}$$

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \Delta\boldsymbol{x}_n \tag{2.3}$$

Then, a sparse linear system $\boldsymbol{A}\Delta\boldsymbol{x} = \boldsymbol{b}$ needs to be solved to yield the update $\boldsymbol{\Delta x}$. When the method converges, $\boldsymbol{x}_n \to \boldsymbol{x}_\infty$, the asymptotic convergence is in general quadratic:

$$\|\boldsymbol{x}_{n+1} - \boldsymbol{x}_\infty\| \leq k\|\boldsymbol{x}_n - \boldsymbol{x}_\infty\|^2. \tag{2.4}$$

Thus, when the initial guess is close enough to the solution, the solver will typically converge in very few steps.

2.1.5 Richardson Extrapolation

In order to achieve higher numerical accuracy, we employ Richardson extrapolation [24] on the resulting steady-state velocity $\mathscr{U}(h)$, as a scalar function of the grid spacing h. This method enables us to detect and analyze nonlinear effects for $\beta \ll 1$, where a higher-order discretization is required because otherwise the numerical error becomes comparable to the difference between the linear analytical regime and the numerical results.

2.2 Algorithm Description

The numerical solver first requires constructing a discrete approximation for the nonlinear PDE system, for a given electric field β and assumed velocity $\mathscr{U}$. The resulting nonlinear discrete system is solved numerically using Newton's Method, so the total force on the particle can be computed. We seek the steady-state velocity $\mathscr{U}$, which corresponds to zero total force, $F(\beta, \mathscr{U}) = 0$, which we solve by the standard secant method.

2.2.1 Spherical Coordinates

Because the system is axisymmetric and the particle is a sphere, we employ spherical coordinates (r, θ, ϕ), as shown in Figure 2.1.

The scalar gradient and the divergence can be written as:

$$\begin{aligned} \boldsymbol{\nabla} f &= \frac{\partial f}{\partial r}\hat{\boldsymbol{r}} + \frac{1}{r}\frac{\partial f}{\partial \theta}\hat{\boldsymbol{\theta}}, && (2.5) \\ \boldsymbol{\nabla} \cdot \boldsymbol{F} &= \frac{1}{r^2}\frac{\partial}{\partial r}\left(F_r r^2\right) + \frac{1}{r\sin\theta}\frac{\partial}{\partial\theta}\left(F_\theta \sin\theta\right). && (2.6) \end{aligned}$$

The scalar Laplacian can be written as:

$$\nabla^2 f = \boldsymbol{\nabla} \cdot (\boldsymbol{\nabla} f) = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \frac{\partial f}{\partial r}\right) + \frac{1}{r^2 \sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta \frac{\partial f}{\partial \theta}\right). \tag{2.7}$$

The vector gradient and vector Laplacian can be written as:

$$\begin{aligned} \boldsymbol{\nabla} \boldsymbol{F} &= \frac{\partial F_r}{\partial r}\hat{\boldsymbol{r}}\hat{\boldsymbol{r}} + \frac{\partial F_\theta}{\partial r}\hat{\boldsymbol{r}}\hat{\boldsymbol{\theta}} + \frac{1}{r}\left(\frac{\partial F_r}{\partial \theta} - F_\theta\right)\hat{\boldsymbol{\theta}}\hat{\boldsymbol{r}} + \frac{1}{r}\left(\frac{\partial F_\theta}{\partial\theta} + F_r\right)\hat{\boldsymbol{\theta}}\hat{\boldsymbol{\theta}}, \\ \hat{\boldsymbol{r}} \cdot \boldsymbol{\nabla}^2 \boldsymbol{F} &= \nabla^2 F_r - \frac{2F_r}{r^2} - \frac{2}{r^2 \sin\theta}\frac{\partial\left(F_\theta \sin\theta\right)}{\partial\theta} \\ \hat{\boldsymbol{\theta}} \cdot \boldsymbol{\nabla}^2 \boldsymbol{F} &= \nabla^2 F_\theta - \frac{F_\theta}{r^2 \sin^2\theta} + \frac{2}{r^2}\frac{\partial F_r}{\partial\theta}. \end{aligned} \tag{2.8}$$

Note that in the spherical coordinate system, the vector Laplacian operator cannot be computed in a decoupled manner as in the Cartesian system, because the unit vectors $\hat{\boldsymbol{r}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}}$ are functions of the coordinates in a curvilinear coordinate system. See Appendix B for the full derivation of the identities above.

2.2.2 Computational Grid

A regular grid of size $n_r \times n_\theta$ is defined, with:

$$\begin{aligned} (r_i, \theta_j) &\in [1, \infty) \times [0, \pi], \\ r_i &= (1 + \Delta_r)^i, \\ \theta_j &= \Delta_\theta \cdot j, \end{aligned} \tag{2.9}$$

where a logarithmic grid spacing has been chosen for r and a uniform grid for θ. Note that $r_0 = 1$ and $R_{max} = (1 + \Delta_r)^{n_r}$, where $\Delta_r = (R_{max})^{1/n_r} - 1$ and $\Delta_\theta = \pi / n_\theta$.

Because the grid is finite, the choice of R_{max} must be large enough so as to have a negligible effect on the solution. On the other hand, in order to have a high resolution grid near the particle surface, Δ_r should be minimized. It should be noted that for a given grid size n_r, these requirements cannot be satisfied simultaneously – thus, a compromise is required between large R_{max} and small Δ_r.

This grid induces a disjoint subdivision of the domain $\Omega = [1, R_{max}] \times [0, \pi] = \bigcup_{ij} \Omega_{ij}$ into cells. A specific cell Ω_{ij} and its center $(\bar{r}_i, \bar{\theta}_j)$ are defined by (2.9):

$$\begin{aligned} \Omega_{ij} &= [r_{i-1}, r_i] \times [\theta_{j-1}, \theta_j], \\ \bar{r}_i &= (r_{i-1} + r_i)/2, \\ \bar{\theta}_j &= (\theta_{j-1} + \theta_j)/2. \end{aligned} \tag{2.10}$$

Each discrete variable is located with respect to its cell as shown at Figure 2.2. $\varPhi$, C and P are represented by their value at the center of each cell, using an all-centered grid, whereas $\boldsymbol{V}$ is represented by its values at cell boundaries, using a staggered grid:

$$\begin{aligned}
\varPhi^h_{[i,j]} &= \varPhi(\bar{r}_i, \bar{\theta}_j), \\
C^h_{[i,j]} &= C(\bar{r}_i, \bar{\theta}_j), \\
P^h_{[i,j]} &= P(\bar{r}_i, \bar{\theta}_j), \\
{V_r^h}_{[i,j]} &= V_r(r_i, \bar{\theta}_j), \\
{V_\theta^h}_{[i,j]} &= V_\theta(\bar{r}_i, \theta_j).
\end{aligned} \tag{2.11}$$

2.2.3 Operator Discretization

A finite-volume method with linear interpolation is used for flux discretization. Define the following discrete central difference operators:

$$\mathcal{D}_r(f^h)_{[i+\frac{1}{2},j]} = \frac{f^h_{[i+1,j]} - f^h_{[i,j]}}{r_{[i+1,j]} - r_{[i,j]}}, \tag{2.12}$$

$$\mathcal{D}_\theta(f^h)_{[i,j+\frac{1}{2}]} = \frac{f^h_{[i,j+1]} - f^h_{[i,j]}}{\theta_{[i,j+1]} - \theta_{[i,j]}}. \tag{2.13}$$

Define the following linear interpolation operators:

$$\mathcal{I}_r(f^h)_{[i+\frac{1}{2},j]} = \frac{r_{[i+\frac{1}{2},j]} - r_{[i,j]}}{r_{[i+1,j]} - r_{[i,j]}} f^h_{[i+1,j]} + \frac{r_{[i+1,j]} - r_{[i+\frac{1}{2},j]}}{r_{[i+1,j]} - r_{[i,j]}} f^h_{[i,j]}, \tag{2.14}$$

$$\mathcal{I}_\theta(f^h)_{[i,j+\frac{1}{2}]} = \frac{r_{[i,j+\frac{1}{2}]} - r_{[i,j]}}{r_{[i,j+1]} - r_{[i,j]}} f^h_{[i,j+1]} + \frac{r_{[i,j+1]} - r_{[i,j+\frac{1}{2}]}}{r_{[i,j+1]} - r_{[i,j]}} f^h_{[i,j]}. \tag{2.15}$$

If the total flux of a cell is equal to zero ($\boldsymbol{\nabla} \cdot \boldsymbol{f} = 0$), then using (2.12), we have

$$\frac{1}{r^2}\mathcal{D}_r\left(f^h_r r^2\right) + \frac{1}{r\sin\theta}\mathcal{D}_\theta\left(f^h_\theta \sin\theta\right) = 0. \tag{2.16}$$

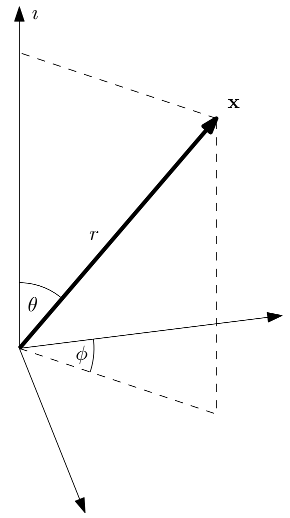

Figure 2.1: Spherical coordinate system is employed for computational grid construction.

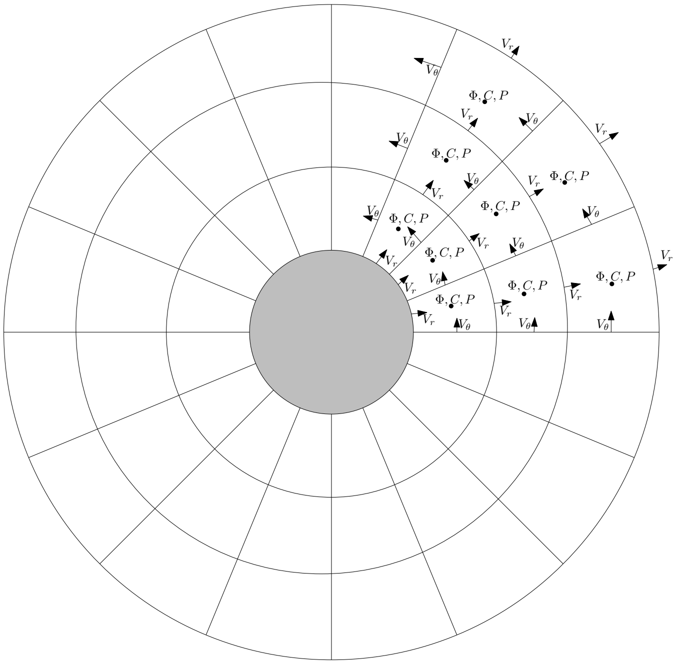

Figure 2.2: The computational grid is defined using a spherical coordinate system, due to axisymmetry of the physical problem and boundary conditions discretization on the particle surface.

2.2.4 System Equations

Charge and salt fluxes (1.13) are discretized on grid cell boundaries (using an upwind scheme $\mathcal{U}$ for numerical stability at large cell Peclet number) using (2.12) and (2.14):

$$\begin{aligned}
I_r^h &= -\mathcal{I}_r(C^h) \cdot \mathcal{D}_r(\Phi^h), \\
I_\theta^h &= -\mathcal{I}_\theta(C^h) \cdot \frac{\mathcal{D}_\theta(\Phi^h)}{r}, \\
J_r^h &= -\mathcal{D}_r(C^h) + \alpha V_r^h \cdot \mathcal{U}_r^{\boldsymbol{V}^h}(C^h), \\
J_\theta^h &= -\frac{\mathcal{D}_\theta(C^h)}{r} + \alpha V_\theta^h \cdot \mathcal{U}_\theta^{\boldsymbol{V}^h}(C^h), \\
\mathcal{U}_r^{\boldsymbol{V}^h}(C^h)[i,j] &= C^h\left[i - \frac{\operatorname{sign}(V_r^h)}{2}, j\right], \\
\mathcal{U}_\theta^{\boldsymbol{V}^h}(C^h)[i,j] &= C^h\left[i, j - \frac{\operatorname{sign}(V_\theta^h)}{2}\right].
\end{aligned} \tag{2.17}$$

Mass flux $\boldsymbol{V}$ is discretized on grid cells boundaries, using the staggered velocity grid. Force components ($\boldsymbol{F} = -\boldsymbol{\nabla} P + \boldsymbol{\nabla}^2 \boldsymbol{V} + \boldsymbol{\nabla}\Phi\nabla^2\Phi$) are discretized on the staggered velocity grid using the finite difference method, where linear interpolation is used for the Coulomb force and for vector Laplacian components.

$$\begin{aligned}
F_r^h &= -\mathcal{D}_r(P^h) + \mathcal{L}(V_r^h) - \frac{2}{r^2}V_r^h - \frac{2}{r^2\sin\theta}\mathcal{I}_r(\mathcal{D}_\theta(V_\theta^h\sin\theta)) + \mathcal{D}_r(\Phi^h)\cdot\mathcal{I}_r(\mathcal{L}(\Phi^h)), \\
F_\theta^h &= -\frac{\mathcal{D}_\theta(P^h)}{r} + \mathcal{L}(V_\theta^h) - \frac{F_\theta^h}{r^2\sin^2\theta} + \frac{2}{r^2}\mathcal{I}_\theta(\mathcal{D}_\theta(F_r^h)) + \frac{\mathcal{D}_\theta(\Phi^h)}{r}\cdot\mathcal{I}_\theta(\mathcal{L}(\Phi^h)), \\
\mathcal{L}(f^h) &= \frac{1}{r^2}\mathcal{D}_r\left(\mathcal{D}_r(f^h)r^2\right) + \frac{1}{r^2\sin\theta}\mathcal{D}_\theta\left(\mathcal{D}_\theta(f^h)\cdot\sin\theta\right).
\end{aligned} \tag{2.18}$$

2.2.5 Boundary Conditions

Far away from the particle boundary (1.10), we have:

$$\mathcal{D}_r\left(\Phi^h\right) = -\beta\cos\bar{\theta}, \quad \mathcal{I}_r\left(C^h\right) = 1 \quad V_r^h = -\mathscr{U}\cos\theta, \quad V_\theta^h = \mathscr{U}\sin\theta. \tag{2.19}$$

For $\theta = 0$ and $\theta = \pi$, the "ghost" points are defined by axial symmetry considerations:

$$\mathcal{D}_\theta\left(\Phi^h\right) = 0, \quad \mathcal{D}_\theta\left(C^h\right) = 0, \quad \mathcal{D}_\theta\left(V_r^h\right) = 0, \quad V_\theta^h = 0 \tag{2.20}$$

The specific boundary conditions at $r = 1$ depend on the chemical properties of particle surface.

Ion-exchanger

The boundary conditions (1.16) are implemented by:

$$\begin{aligned}
0 &= \mathcal{I}_r(\Phi^h + \log C^h), \\
0 &= \mathcal{D}_r(\Phi^h - \log C^h), \\
V_r^h &= 0, \\
V_\theta^h &= 4 \log\left(\tfrac{1}{2}\left(1 + \exp\left\{\mathcal{I}_\theta(\zeta^h)/2\right\}\right)\right) \cdot \mathcal{D}_\theta(\zeta^h), \\
\zeta^h &= -\log\gamma - \mathcal{I}_r(\Phi^h).
\end{aligned} \tag{2.21}$$

Highly charged inert particle

The boundary conditions (1.17) are implemented by:

$$\begin{aligned}
0 &= \mathcal{D}_r(C^h) - \mathrm{Du} \cdot \left(\mathcal{D}_\theta \left(\sin\theta \cdot \mathcal{D}_\theta \mathcal{I}_r \left(\Phi^h - \log C\right)\right)\right) / \sin\theta, \\
0 &= \mathcal{D}_r(C^h) + \mathcal{I}_r(C^h)\mathcal{D}_r(\Phi^h), \\
V_r^h &= 0, \\
V_\theta^h &= \mathcal{I}_\theta(\zeta^h) \cdot \mathcal{D}_\theta(\mathcal{I}_r\Phi^h - \log \mathcal{I}_r C^h) + 4 \log 2 \cdot \mathcal{D}_\theta(\log \mathcal{I}_r C^h), \\
\zeta^h &= \bar{\zeta} - \log \mathcal{I}_r C.
\end{aligned} \tag{2.22}$$

2.3 Solver Design

Let $\mathcal{O} : \mathbb{R}^N \to \mathbb{R}^N$ be a nonlinear operator. In order to solve the equation $\mathcal{O}(\boldsymbol{x}) = \mathbf{0}$, Newton's method is applied, given an initial solution $\boldsymbol{x}_0$. Each step requires the computation of the residual vector, $\boldsymbol{r} = \mathcal{O}(\boldsymbol{x})$, the gradient matrix of the operator, $\boldsymbol{G} = \boldsymbol{\nabla}\mathcal{O}(\boldsymbol{x})$ (given the current solution $\boldsymbol{x}$) and the solution of the sparse linear system $\boldsymbol{G}\Delta\boldsymbol{x} = -\boldsymbol{r}$.

2.3.1 Operator Representation

The system variable $\boldsymbol{x}$ is defined as the concatenation $\boldsymbol{x} = [\Phi, C, V_r, V_\theta, P]$. Thus, each problem variable can be computed by applying an appropriate projection operator on $\boldsymbol{x}$. The system can be written as a nonlinear operator $\mathcal{O}(\boldsymbol{x}) = \mathbf{0}$ acting on the system variable $\boldsymbol{x}$, where $\mathcal{O}$ is the system equations operator, derived by concatenation of the system equations (described in Subsection 2.2.4) and the boundary conditions operators (described in Subsection 2.2.5). This way, the same numerical solver handles both the nonlinear system equations and nonlinear boundary conditions.

2.3.2 Steady-state

Let β be a given nondimensional electric field. Assume that $\mathscr{U}\hat{\boldsymbol{\imath}}$ is the drift velocity of the particle (given that the fluid is at rest far away from it). After the convergence of the solver, once the residual norm is below a prescribed threshold $\|\mathcal{O}(\boldsymbol{x}_n)\| < \epsilon$, the total force acting on the particle can be computed by integrating $\mathsf{T} \cdot \hat{\boldsymbol{n}}$ over the particle surface.

Due to symmetry considerations, the total force $\boldsymbol{F}$ is aligned with $\hat{\boldsymbol{\imath}}$ and it vanishes iff $\hat{\boldsymbol{\imath}} \cdot \boldsymbol{F} = 0$.

$$\begin{aligned} F_\imath = \hat{\boldsymbol{\imath}} \cdot \boldsymbol{F} &= \oint_{\mathcal{S}} (\hat{\boldsymbol{\imath}} \cdot \mathsf{T} \cdot \hat{\boldsymbol{n}})\, dA = \int_0^\pi f_\imath(\theta) 2\pi \sin\theta d\theta, \\ f_\imath = \hat{\boldsymbol{\imath}} \cdot \mathsf{T} \cdot \hat{\boldsymbol{r}} &= \left(-P + 2\mathcal{D}_r(V_r) + \frac{1}{2}\left(\mathcal{D}_r(\varPhi)\right)^2 - \frac{1}{2r^2}\left(\mathcal{D}_\theta(\varPhi)\right)^2 \right) \cos\theta \\ &\quad - \left(\mathcal{D}_r(V_\theta) - \frac{V_\theta}{r} + \frac{1}{r}\mathcal{D}_r(\varPhi)\mathcal{D}_\theta(\varPhi) \right) \sin\theta. \end{aligned} \tag{2.23}$$

The integral above is approximated by 1D numerical quadrature (using the mid-point rule)

$$F_\imath^h(\beta, \mathscr{U}^h) = \sum_{i=1}^{n_\theta} f^h(\bar{\theta}_i) \cdot 2\pi \sin\bar{\theta}_i \cdot \Delta\theta_i, \tag{2.24}$$

to yield the total force $F_\imath^h(\beta, \mathscr{U}^h)$ as a function of the drift velocity $\mathscr{U}^h$. Since the goal is to find the steady-state solution, $F_\imath^h$ is required to be zero – and the appropriate $\mathscr{U}^h$ is found by a simple 1D root-finding algorithm, applied as an outer loop:

$$F_\imath^h(\beta, \mathscr{U}^h(\beta)) = 0 \tag{2.25}$$

2.3.3 Continuation

The iterative solver above can in principle be used to compute the steady-state solution for any given β. For $\beta \ll 1$, the linear terms are the dominant ones, so the solution is approximately linear in β (as derived in [1]), and the solver convergence is fast. However, this is no longer true for $\beta \gtrsim 1$, because the nonlinear terms become dominant and the iterative solver may not converge at all if we start with an arbitrary initial guess. In order to find a solution for such β, a continuation method is used: the solver is applied to a sequence of $\{\beta_i\}_{i=0}^n$ such that $\beta_0 = 0$, $\beta_n = \beta$ and the solution $\boldsymbol{x}_i$ for β_i is used as the solver initializer for the problem of β_{i+1}.

2.4 Solver Implementation

Object-Oriented Design methodology is used for implementation. The MATLAB programming language is chosen due to strong numerical capabilities and high-level language features. The source code of the solver is provided in [25].

MATLAB direct solver is applied to solve the sparse linear system for each Newton step, using the backslash operator. The library used for solving the linear system is UMFPACK 5.0 [26] (Unsymmetric MultiFrontal Sparse LU Factorization Package), as reported by MATLAB function call, `spparms('spumoni', 2)`.

2.4.1 Operator Interface

The base `Operator` interface is defined by supporting the computation of a residual vector `Operator.res()` and a gradient matrix `Operator.grad()`, given an input vector $\boldsymbol{x}$. This interface is implemented by specific operator classes, which are used to construct the system operator $\mathcal{O}$, so that the Newton solver (that requires computing the residual and the gradient) can be applied automatically. Each such operator may have other operators as its inputs, so the residual and the gradient are computed recursively, using the chain rule (applied to operators).

Linear operator

If $\mathcal{O}$ is linear, it can be represented by a matrix L, such that:

$$\mathcal{O}(\boldsymbol{x}) = L\boldsymbol{x}, \quad \boldsymbol{\nabla}\mathcal{O} = L. \tag{2.26}$$

Note that finite difference $\mathcal{D}$, interpolation $\mathcal{I}$ and upwind selection $\mathcal{U}$ operators can be implemented as linear sparse operators, by constructing an appropriate sparse matrix L, having $O(\dim \boldsymbol{x})$ non-zeroes. This way, the computation of the residual is very fast, taking $O(\dim \boldsymbol{x})$ time, and the gradient does not depend on $\boldsymbol{x}$.

Pointwise scalar function

Let $f : \mathbb{R} \to \mathbb{R}$ be a differentiable 1D function, whose derivative is denoted by $f' : \mathbb{R} \to \mathbb{R}$. The operator $\mathcal{F}$ can be defined to represent a pointwise application of f:

$$\mathcal{F}(\boldsymbol{x}) = [f(x_1); \ldots; f(x_n)], \quad \boldsymbol{\nabla}\mathcal{F}(\boldsymbol{x}) = \operatorname{diag}\{f'(x_1), \ldots, f'(x_n)\}. \tag{2.27}$$

f is defined and differentiated automatically using the MATLAB symbolic toolbox.

Constant value

A constant operator $\mathcal{C}$ is defined by the constant vector $\boldsymbol{c}$ (which does not depend on $\boldsymbol{x}$):

$$\mathcal{C}(\boldsymbol{x}) = \boldsymbol{c}, \quad \boldsymbol{\nabla}\mathcal{C} = \boldsymbol{0}. \tag{2.28}$$

Binary operators

Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be two operators, so that their pointwise addition, subtraction, and multiplication, are defined as follows:

$$\begin{aligned} (\mathcal{O}_1 \pm \mathcal{O}_2)(\boldsymbol{x}) &= \mathcal{O}_1(\boldsymbol{x}) \pm \mathcal{O}_2(\boldsymbol{x}), \qquad (2.29)\\ \boldsymbol{\nabla}(\mathcal{O}_1 \pm \mathcal{O}_2)(\boldsymbol{x}) &= \boldsymbol{\nabla}\mathcal{O}_1(\boldsymbol{x}) \pm \boldsymbol{\nabla}\mathcal{O}_2(\boldsymbol{x}), \\ (\mathcal{O}_1 \cdot \mathcal{O}_2)(\boldsymbol{x}) &= \mathcal{O}_1(\boldsymbol{x}) \cdot \mathcal{O}_2(\boldsymbol{x}), \qquad (2.30)\\ \boldsymbol{\nabla}(\mathcal{O}_1 \cdot \mathcal{O}_2)(\boldsymbol{x}) &= \operatorname{diag}(\mathcal{O}_2(\boldsymbol{x}))\boldsymbol{\nabla}\mathcal{O}_1(\boldsymbol{x}) + \operatorname{diag}(\mathcal{O}_1(\boldsymbol{x}))\boldsymbol{\nabla}\mathcal{O}_2(\boldsymbol{x}). \end{aligned}$$

N-ary operators

Given N operators $\{\mathcal{O}_i\}_{i=1}^N$, their concatenation $\mathcal{O}$ is defined as:

$$\begin{aligned} \mathcal{O}(\boldsymbol{x}) &= [\mathcal{O}_1(\boldsymbol{x}); \ldots ; \mathcal{O}_N(\boldsymbol{x})], \\ (\boldsymbol{\nabla}\mathcal{O})(\boldsymbol{x}) &= [(\boldsymbol{\nabla}\mathcal{O}_1)(\boldsymbol{x}); \ldots ; (\boldsymbol{\nabla}\mathcal{O}_N)(\boldsymbol{x})]. \end{aligned} \tag{2.31}$$

2.4.2 Handling the Singularity

The electric potential $\mathit{\Phi}$ (in electrophoresis case) and the pressure P (in both cases) are defined up to an additive constant, since the system equations and the boundary conditions contain only their derivatives. In order to overcome the singularity of the linear system $\boldsymbol{G}\Delta\boldsymbol{x} = -\boldsymbol{r}$, we set the variable at a specific grid point to be 0. In order to retain a square system, we remove one conservation equation at this grid point too. In the case of pressure P, we remove one mass conservation equation, $\boldsymbol{\nabla} \cdot \boldsymbol{V} = 0$, at this grid point. In the case of electric potential, we remove the charge conservation equation, $\boldsymbol{\nabla} \cdot (C\boldsymbol{\nabla}\mathit{\Phi}) = 0$, at this grid point. For $\boldsymbol{r} = \mathcal{O}(\boldsymbol{x})$ and $\boldsymbol{G} = \boldsymbol{\nabla}\mathcal{O}(\boldsymbol{x})$, the new square linear system reads:

$$\mathbb{R}\boldsymbol{G}\mathbb{P} \cdot \delta\boldsymbol{x} = -\mathbb{R}\boldsymbol{r}, \tag{2.32}$$

where $\mathbb{P}$ matrix is used to remove the columns and $\mathbb{R}$ matrix is used to remove the rows of $\boldsymbol{G}$, where the update step is $\Delta\boldsymbol{x} = \mathbb{P} \cdot \delta\boldsymbol{x}$.

2.4.3 Iterative Newton solver

Given an initial solution $\boldsymbol{x}_0$ and prescribed values for β and $\mathscr{U}$, we apply the following algorithm until convergence ($\|\boldsymbol{r_n}\| < \epsilon$):

1. Update the system operator: $\texttt{op} = \texttt{System}(\boldsymbol{x}_n)$
2. Compute the gradient: $\boldsymbol{G}_n = \texttt{op.grad()}$
3. Compute the right-hand side: $\boldsymbol{r}_n = \texttt{op.res()}$
4. Solve sparse linear system: $\mathbb{R}\boldsymbol{G}_n\mathbb{P}\Delta\boldsymbol{x}_n = -\mathbb{R}\boldsymbol{r}_n$
5. Update solution: $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \mathbb{P}\Delta\boldsymbol{x}_n$

2.5 Source code overview

In the following section we present the source code of the numerical framework and demonstrate its usage for solving a boundary-value problem.

2.5.1 Variable

We define the variable vector $\boldsymbol{x}$ as `Variable` operator (Figure 2.3).

```
classdef Variable < handle
properties
    value;
end
methods
    function self = Variable(varargin)
        self.value = col(varargin{:});
    end
    function r = res(self)
        r = self.value(:);
    end
    function G = grad(self)
        G = speye(numel(self.value));
    end
    function update(self, dv)
        self.value = self.value + dv;
    end
end
```

Figure 2.3: Variable class source code.

Then the initial solution for the solver is set using $\boldsymbol{x} =$ `Variable`$(\varPhi, C, V_r, V_t, P)$ from the initial conditions of the solver.

2.5.2 Grid

Most of the operators are using a `Grid` object (Figure 2.4), enabling validation of operators compatibility and convenient interpolation between grids. Note that each operator must define its `res()` and `grad()` functions, for residual and gradient computation, respectively. An `Operator` interface is used (Figure 2.5) to support operator overloading MATLAB syntax.

Thus, various computations can be applied on simple operators, creating more complex operators, such as `op4 = op1 * exp(op2 + tanh(op3))`. Since each operator can compute its residual and gradient, we can compute automatically the residual and gradient of `op4` from these of `op1`, `op2`, `op3` using the derivation rules described at 2.4.1.

2.5.3 Operators

An operator can be created with n existing operators as input. `Const` is an operator that corresponds to $n = 0$, whose residual value is constant and gradient matrix is zero (see Figure 2.6).

```
% Regular structured grid class
classdef Grid < handle
properties
    x, y;
    X, Y;
    size;
    numel;
end
methods
    function self = Grid(x, y)
        self.x = x(:);
        self.y = y(:);
        [self.X, self.Y] = ndgrid(self.x, self.y);
        self.size = [numel(self.x), numel(self.y)];
        self.numel = prod(self.size);
    end
end
```

Figure 2.4: Grid class source code.

Unary operators

Unary operators have one existing operator as input. `Func` operator implements point-wise analytic function application (whose derivative is computed using the symbolic differentiation toolbox) on the same grid of the input operator (Figure 2.7). `Linear` operator is implemented using multiplication by a constant sparse matrix `L`, which is also the gradient of the operator (Figure 2.8). `Deriv` and `Interp` operators (implementing discrete differentiation and grid interpolation) derive from `Linear` base class operator and specify `L` sparse matrix accordingly.

Binary operators

Binary operators such as addition, subtraction and point-wise multiplication are implemented as operators whose output grid is the same as the grid of their inputs. The implementation is used by operator overloading in the base class `Operator`.

N-ary operators

All equations and boundary condition operators are defined as operators `{op1, op2, ... opN}`. Then, they are combined using the N-ary operator `op = Join(op1, op2, ... opN)`, creating the system operator `op` (Figure 2.9).

2.5.4 Sample source code

The following example shows how to define the following boundary-value problem:

$$\nabla^2 F(x,y) = 1 \qquad (x,y) \in [0,1]^2 \tag{2.33}$$

$$F(0,y) = F(1,y) = 0 \qquad y \in [0,1] \tag{2.34}$$

$$\frac{\partial F}{\partial y}(x,0) = \frac{\partial F}{\partial y}(x,1) = 0 \qquad x \in [0,1] \tag{2.35}$$

and solve it using one Newton method step. The source code is presented at Figure 2.10.

```matlab
classdef Operator < handle
properties
    grid; % a grid object for the output
end
methods
    function op = Operator(grid)
        op.grid = grid;
    end
    function op = plus(op1, op2)
        op = Sum(op1, op2);
    end
    function op = minus(op1, op2)
        op = Minus(op1, op2);
    end
    function op = uminus(op)
        op = Product(-1, op);
    end
    function op = mtimes(op1, op2)
        op = Product(op1, op2);
    end
    function op = log(op)
        op = Func(op, 'log(x)');
    end
    function op = exp(op)
        op = Func(op, 'exp(x)');
    end
    function op = sinh(op)
        op = Func(op, 'sinh(x)');
    end
    function op = cosh(op)
        op = Func(op, 'cosh(x)');
    end
    function op = tanh(op)
        op = Func(op, 'tanh(x)');
    end
end
```

Figure 2.5: Operator interface source code.

```
classdef Const < Operator
properties
    val;
end
methods
    function self = Const(grid, val)
        self = self@Operator(grid);
        if isa(val, ’float’)
            if numel(val) == 1
                val = repmat(val, grid.size);
            end
            self.val = val;
        else % function handle evaluated point-wise
            if isa(val, ’char’)
                val = str2func([’@(r,t) ’ val]);
            end
            self.val = arrayfun(val, grid.R, grid.T);
        end
    end
    function r = res(self)
        r = self.val(:);
    end
    function G = grad(self)
        G = sparse(0); % [0]
    end
end
```

Figure 2.6: Constant operator source code.

```
classdef Func < Operator
properties
    op; % Input operator
    func; % y = f(x)
    deriv; % dy/dx = f’(x)
end
methods
    function self = Func(op, func)
        grid = op.grid; % Reuse operator’s grid
        self = self@Operator(grid);
        self.op = op;
        if isa(func, ’char’)
                func = sym(func);
        end
        self.func = matlabFunction(func);
        self.deriv = matlabFunction(diff(func));
    end
    function r = res(self)
        r = self.func( self.op.res() );
    end
    function G = grad(self)
        r = self.op.res();
        G = spdiag( self.deriv(r) ) * self.op.grad();
    end
end
```

Figure 2.7: Pointwise function source code.

```
classdef Linear < Operator
properties
    L;
    op;
end
methods
    function self = Linear(grid, op, L)
        self = self@Operator(grid);
        self.op = op;
        self.L = L;
    end
    function r = res(self)
        r = self.L * self.op.res();
    end
    function G = grad(self)
        G = self.L * self.op.grad();
    end
end
```

Figure 2.8: Linear operator source code.

```
classdef Join < handle
properties
    ops;
end
methods
    function self = Join(operators)
        self.ops = operators;
        for k = 1:numel(self.ops)
                op = self.ops{k};
        end
    end
    function r = res(self)
        r = cell(numel(self.ops), 1);
        for k = 1:numel(self.ops)
            r{k} = self.ops{k}.res();
        end
        r = cat(1, r{:});
    end
    function G = grad(self)
        G = cell(numel(self.ops), 1);
        for k = 1:numel(self.ops)
            G{k} = self.ops{k}.grad();
        end
        G = cat(1, G{:});
    end
end
```

Figure 2.9: Join operator source code.

```
x = linspace(0, 1);
y = linspace(0, 1);
g = Grid(x, y); % Grid definition
x_ = (x(1:end-1) + x(2:end)) * 0.5;
y_ = (y(1:end-1) + y(2:end)) * 0.5;
xi = x(2:end-1);
yi = y(2:end-1);
v = Variable(zeros(g.size));
F = Linear(g, v, speye(g.numel)); % Variable Definition
Fx = Deriv(Grid(x_, y), F, 1);
Fy = Deriv(Grid(x, y_), F, 2);
Fxx = Deriv(Grid(xi, yi), Fx, 1);
Fyy = Deriv(Grid(xi, yi), Fy, 2);
eq = {(Fxx + Fyy) - Const(1)};               % Laplacian(F) = 1
bnd = {Interp(Grid(0.0, y(2:end-1)), F); ... % F(0, y) = 0
           Interp(Grid(1.0, y(2:end-1)), F); ... % F(1, y) = 0
       Interp(Grid(x, 0.0), Fy); ...         % dF/dy(x, 0) = 0
       Interp(Grid(x, 1.0), Fy)}                     % dF/dy(x, 1) = 0
op = Join([eq; bnd]); % System operator
r = op.res()
G = op.grad()
v.update(-(G\r)) % Solve by Newton iteration
```

Figure 2.10: Sample problem source code.

Chapter 3

Results and Discussion

3.1 Asymptotic Analysis

In order to validate the numerical results, an asymptotic analysis has been applied to the nonlinear systems of ion-exchanger migration and highly charged particle electrophoresis. We present here the asymptotic analysis for the ion-exchanger case [1], while the asymptotic analysis for the electrophoresis case [2] is derived by Schnitzer and Yariv in [27].

The derivation is performed as follows. The nonlinear system (composed of the governing equations and the boundary conditions) is written as $\mathcal{O}(\boldsymbol{x}; \beta) = \mathbf{0}$. For small β, the solution to the nonlinear system can be expanded in an asymptotic series in β:

$$\boldsymbol{x} = \boldsymbol{x}(\beta) \approx \sum_n \boldsymbol{x}_n \beta^n. \tag{3.1}$$

The nonlinear terms are expanded around $\beta = 0$:

$$\mathbf{0} = \mathcal{O}(\boldsymbol{x}) = \sum_i \mathcal{O}_i(\boldsymbol{x}_0, \ldots \boldsymbol{x}_i)\beta^i. \tag{3.2}$$

Thus, the $O(\beta^k)$ term $\boldsymbol{x}_k$ can be found recursively by solving $\mathcal{O}_k(\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) = \mathbf{0}$, given the previous solutions for $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{k-1}$. The derivation is validated using the MATLAB symbolic toolbox by the code at `symbolic/asymp.m` [25].

3.1.1 First-Order (linear in β) Solution

For $\beta = 0$ (no electric field is applied), the steady-state solution is $\mathscr{U} = 0$:

$$\varPhi_0(r, \theta) = 0, \quad C_0(r, \theta) = 1, \quad \boldsymbol{V}_0(r, \theta) = \mathbf{0}, \quad P_0(r, \theta) = 0. \tag{3.3}$$

Assuming $\beta \ll 1$, the linearized equations and boundary conditions are

$$\nabla^2 \varPhi_1 = 0, \quad \nabla^2 C_1 = 0, \quad \boldsymbol{\nabla}^2 \boldsymbol{V}_1 - \boldsymbol{\nabla} P_1 = \mathbf{0}. \tag{3.4}$$

The first-order terms $\boldsymbol{x}_1$ are given by (see [1] for full derivation)

$$\varPhi_1 = \left(\tfrac{1}{4r^2} - r\right)\cos\theta, \quad C_1 = \tfrac{3}{4r^2}\cos\theta, \quad \Psi_1 = \mathscr{U}_1\left(\tfrac{1}{r} - r^2\right)\tfrac{\sin^2\theta}{2}, \quad P_1 = 0, \tag{3.5}$$

and the linear velocity term is:

$$\boldsymbol{V}_1 = \mathscr{U}_1\left(\hat{\boldsymbol{r}}\left(-1 + \frac{1}{r^3}\right)\cos\theta + \hat{\boldsymbol{\theta}}\left(1 + \frac{1}{2r^3}\right)\sin\theta\right) \tag{3.6}$$

$$\mathscr{U}_1(\gamma) = 2\log\left(\frac{1+\gamma^{-\frac{1}{2}}}{2}\right). \tag{3.7}$$

For $\gamma > 1$, the Debye layer has a positive charge, corresponding to a negative particle charge – yielding negative drift velocity, $\mathscr{U} < 0$. For $\gamma < 1$, we have $\mathscr{U} > 0$.

3.1.2 Second-Order (quadratic in β) Solution

The quadratic terms $\boldsymbol{x}_2$ are computed by solving a linear PDE system with a right-hand side determined by the linear terms $\boldsymbol{x}_1$, yielding

$$\varPhi_2 = \left(\frac{\mathscr{U}_1\,\alpha}{32} - \frac{1}{16}\right)\frac{3\cos^2\theta - 1}{r^3} - \frac{3 + 3\sin^2\theta\,\left(4\,r^3 - 1\right)}{32\,r^4} - \frac{3\,\mathscr{U}_1\,\alpha - 6}{32\,r}, \tag{3.8}$$

$$C_2 = \left(\frac{5\,\mathscr{U}_1\,\alpha}{32} + \frac{1}{16}\right)\frac{3\cos^2\theta - 1}{r^3} - \frac{3\,\mathscr{U}_1\,\alpha - 6}{32\,r} + \frac{3\,\mathscr{U}_1\,\alpha\,\left(2\,r^3\sin^2\theta + \sin^2\theta - 1\right)}{16\,r^4}, \tag{3.9}$$

$$\Psi_2 = \left(\frac{9}{16(\sqrt{\gamma}+1)} - \frac{3}{16}\mathscr{U}_1(\mathscr{U}_1\alpha + 1)\right)\left(\frac{1}{r^2} - 1\right)\sin^2\theta\cos\theta. \tag{3.10}$$

Note that the quadratic term does not contribute to the total force and steady-state velocity terms, but it does change the fluid flow, the electric potential and the ionic concentration.

3.1.3 Third-Order (cubic in β) Solution

Note that, due to symmetry considerations, $\mathscr{U}(\beta)$ is an anti-symmetric function. Therefore, $\mathscr{U}_2 = 0$ and the next velocity term is the cubic one:

$$\mathscr{U}(\beta) \approx \beta\mathscr{U}_1 + \beta^3\mathscr{U}_3 + O(\beta^5). \tag{3.11}$$

The cubic terms $\boldsymbol{x}_3$ are computed by solving a linear PDE system with a right-hand side determined by the linear terms $\boldsymbol{x}_1$ and the quadratic terms $\boldsymbol{x}_2$.

$$
\begin{aligned}
\varPhi_3 = {} & \frac{15\mathscr{U}_1\alpha+6}{64}\cos\theta - \frac{3\mathscr{U}_1\alpha}{32}\cos^3\theta + \frac{183{\mathscr{U}_1}^2\alpha^2-839\mathscr{U}_1\alpha-470}{2560r^2}\cos\theta \\
& + \frac{3\cos\theta\left(5\mathscr{U}_1\alpha-6\cos^2\theta-4\mathscr{U}_1\alpha\cos^2\theta+4\right)}{64r^3} + \frac{\left(2\mathscr{U}_1\alpha-1\right)\left(\cos\theta-3\cos^3\theta\right)}{64r^5} \\
& + \cos^3\theta\left(\frac{15\mathscr{U}_1\alpha}{64}+\frac{3}{32}\right)r^{-2} - \left(3\cos\theta-5\cos^3\theta\right)\left(-\frac{19{\mathscr{U}_1}^2\alpha^2}{5120}+\frac{97\mathscr{U}_1\alpha}{5120}+\frac{3\mathscr{U}_2\alpha}{320}+\frac{21}{1280}\right)r^{-4} \\
& + \cos^3\theta\left(\frac{\mathscr{U}_1\alpha}{64}+\frac{3}{64}\right)r^{-6},
\end{aligned}
$$

$$
\begin{aligned}
C_3 = {} & \frac{3{\mathscr{U}_1}^2\alpha^2\cos^3\theta}{32} + \frac{\cos\theta\left(797{\mathscr{U}_1}^2\alpha^2+419\mathscr{U}_1\alpha-512\mathscr{U}_2\alpha+174\right)}{2560r^2} - \frac{3\mathscr{U}_1\alpha\cos\theta\left(5\mathscr{U}_1\alpha+2\right)}{64} \\
& - \left(3\cos\theta-5\cos^3\theta\right)\left(\frac{59{\mathscr{U}_1}^2\alpha^2}{5120}+\frac{103\mathscr{U}_1\alpha}{5120}+\frac{69\mathscr{U}_2\alpha}{320}+\frac{3}{1280}\right)r^{-4} \\
& + \frac{\alpha\left(\cos\theta-3\cos^3\theta\right)\left(5\alpha{\mathscr{U}_1}^2+2\mathscr{U}_1+16\mathscr{U}_2\right)}{128r^5} \\
& - \frac{3\alpha\cos\theta\left(-8\alpha{\mathscr{U}_1}^2\cos^2\theta+7\alpha{\mathscr{U}_1}^2+2\mathscr{U}_1+32\mathscr{U}_2\cos^2\theta-32\mathscr{U}_2\right)}{128r^3} \\
& + \frac{{\mathscr{U}_1}^2\alpha^2\cos^3\theta}{32r^6} - \frac{3\mathscr{U}_1\alpha\cos^3\theta\left(5\mathscr{U}_1\alpha+2\right)}{64r^2},
\end{aligned}
$$

$$
\begin{aligned}
\Psi_3 = {} & r^2\sin^2\theta\left(\frac{\mathscr{W}_2}{15}-\frac{\mathscr{W}_1}{3}+\frac{209}{3360}\right) - \sin^2\theta\left(\frac{5\mathscr{W}_2}{3}-\frac{\mathscr{W}_1}{3}+\frac{35}{264}\right)r^{-1} + \sin^4\theta\left(2\mathscr{W}_2+\frac{761}{5632}\right)r^{-1} \\
& - \frac{\sin^2\theta\left(1848r^6\sin^2\theta-2079r^3\sin^2\theta+924r^3-7\sin^2\theta+10\right)}{19712r^5}\,\frac{4\sin^2\theta-5\sin^4\theta}{r^3} \\
& + \left(\frac{\mathscr{U}_1\alpha}{16}-\frac{1}{8}\right)\frac{2r^3-3r^2+1}{4r}\sin^2\theta,
\end{aligned}
$$

$$
\begin{aligned}
\mathscr{W}_1 = {} & \frac{69}{512\left(\sqrt{\gamma}+1\right)} - \frac{1407\alpha^2\log\left(\frac{\sqrt{\gamma}+1}{2\sqrt{\gamma}}\right)^3}{1280} - \frac{123\log\left(\frac{\sqrt{\gamma}+1}{2\sqrt{\gamma}}\right)}{1280} - \frac{27}{512\left(\sqrt{\gamma}+1\right)^2} \\
& - \frac{1899\alpha\log\left(\frac{\sqrt{\gamma}+1}{2\sqrt{\gamma}}\right)^2}{2560} - \frac{\alpha\log\left(\frac{\sqrt{\gamma}+1}{2\sqrt{\gamma}}\right)\left(\frac{81}{16\left(\sqrt{\gamma}+1\right)}-\log\left(\frac{\sqrt{\gamma}+1}{2\sqrt{\gamma}}\right)\left(\frac{27\alpha\log\left(\frac{\sqrt{\gamma}+1}{2\sqrt{\gamma}}\right)}{4}+\frac{27}{8}\right)\right)}{320} \\
& + \frac{21\alpha\log\left(\frac{\sqrt{\gamma}+1}{2\sqrt{\gamma}}\right)}{128\left(\sqrt{\gamma}+1\right)},
\end{aligned}
$$

$$\begin{aligned}\mathscr{W}_2 = {} & \frac{9\log\left(\frac{\sqrt{\gamma}+1}{2\sqrt{\gamma}}\right)}{256} + \frac{21\alpha^2\log\left(\frac{\sqrt{\gamma}+1}{2\sqrt{\gamma}}\right)^3 + 15\alpha\log\left(\frac{\sqrt{\gamma}+1}{2\sqrt{\gamma}}\right)^2}{64} - \frac{27}{512\left(\sqrt{\gamma}+1\right)^2} \\ & - \frac{297\alpha\log\left(\frac{\sqrt{\gamma}+1}{2\sqrt{\gamma}}\right)+54}{1024\left(\sqrt{\gamma}+1\right)}.\end{aligned}$$

The cubic solution satisfies the boundary conditions only when $\alpha\mathscr{U}_1 = 0$, due to boundary conditions mismatch for C at $r \to \infty$, when the diffusion and the advection terms become of the same order of magnitude. The cubic correction velocity terms (for $\alpha = 0$) turn out to be:

$$\mathscr{U}_3(\gamma) = \frac{31}{320(\sqrt{\gamma}+1)} - \frac{9}{320(\sqrt{\gamma}+1)^2} + \frac{1}{1680} - \frac{11}{160}\log\left(\frac{1+\gamma^{-\frac{1}{2}}}{2}\right). \tag{3.12}$$

The cubic term becomes dominant for small β when $\gamma \approx 1$. Note that, for $\gamma = 1$, the linear term vanishes and the cubic term of the velocity is $\mathscr{U}(\beta) = \frac{1129}{26880}\beta^3$.

Observe that a $\mathscr{U} = 0$ solution may exist for a critical $\beta_c > 0$ and $\alpha = 0$, if it satisfies:

$$\beta_c^2 = -\frac{\mathscr{U}_1(\gamma)}{\mathscr{U}_3(\gamma)}. \tag{3.13}$$

Specifically, for $\gamma = 1 + \epsilon$, where $0 < \epsilon \ll 1$, $\beta_c = O\left(\sqrt{\epsilon}\right)$:

$$\beta_c \approx \sqrt{-\mathscr{U}_1'(1)\epsilon/\mathscr{U}_3(1)} = \sqrt{13440\epsilon/1129} \approx 3.45\sqrt{\epsilon}. \tag{3.14}$$

That is, there may be a specific nonzero external field for which the particle remains at rest. The physical viability of this solution has yet to be explored.

3.2 Numerical Results

The numerical solver is applied to the nonlinear problem of an ion-exchanger migration driven by an electric field [1] and the electrophoresis of highly charged surface inert particle [2]. The solver is applied to a series of values of the electric field β, ranging from $\beta \ll 1$ to $\beta \sim 1$. For each β value, the Newton solver is initialized with the linear solution and is iterated K times. Secant method iteration is applied M times to find the steady-state velocity by finding $\mathscr{U}$ such that $F_\imath(\mathscr{U},\beta) = 0$, thus computing the steady-state velocity $\mathscr{U}(\beta)$. The resulting $\mathscr{U}(\beta)$ is plotted on a log-log scale, so that the asymptotic behaviour of the solution may be examined for a wide range of β values.

In order to validate the numerical solver, we employ the results of the asymptotic analysis of the nonlinear system. The numerical results are compared to the asymptotic solutions, and a good correspondence is observed.

3.2.1 Ion-exchanger results

For $\beta \ll 1$, the linear regime [5] is dominant. However, when $\gamma \approx 1$, the cubic regime, where $\mathscr{U} = O(\beta^3)$, dominates the linear one, even for weak electric fields. This phenomenon is confirmed both by the numerical results and by asymptotic expansion of the nonlinear system (for $\alpha = 0$). The switch from the linear regime to the cubic regime occurs at an electric field value of $\beta = O(\sqrt{|\gamma - 1|})$. The numerical results $\mathscr{U}^h$, as well as the analytical ones (3.11), are given in Figure 3.1. The numerical solver uses $N_r \times N_\theta = (400 \times 50)$, $R_{\max} = 10^7$ ($\Delta_r \approx 0.041$), $K = 3$, $M = 5$ and $\beta = 10^t$ for $t = -2 : 0.1 : 0.5$.

The linear solution for $\gamma \approx 1$, yields a sign change of the zeta potential $\zeta = \log(C/\gamma)$, when $|\log \gamma| \approx 3\beta/4$. This results in a sign change in the slip velocity, which can be approximated for small $|\zeta| \ll 1$ as $V_\theta \approx \zeta \cdot \boldsymbol{\nabla}_S \Phi$. This sign change creates a vortex at the downstream end of the ion-exchanger, as demonstrated by the numerical solution and the asymptotic analysis results in Figures 3.2 and 3.3. Note that there is a good correspondence between the numerical results and the analytical solution for $\beta < 1$, validating the vortex formation.

3.2.2 Electrophoresis results

For $\beta \ll 1$, the linear regime [2] is dominant. In order to verify the solver, the numerical steady-state velocity is compared to the analytical solution [27], derived by Schnitzer and Yariv. Since the cubic correction is much smaller than the linear solution for $\beta \ll 1$, the difference between numerical results and the linear solution is compared to the cubic correction term.

Because the leading order of the solution $\mathscr{U}(\beta)$ is linear for $\beta \ll 1$, the numerical discretization error is $O(\beta)$ as well. In order to reduce the discretization error of the numerical solution (2.25) and after verifying that the discretization error converges as $O(h^2)$ (Figure 3.4), we employ Richardson extrapolation to $\mathscr{U}^h$ and $\mathscr{U}^{2h}$:

$$\hat{\mathscr{U}} = \frac{4}{3}\mathscr{U}^h - \frac{1}{3}\mathscr{U}^{2h}. \tag{3.15}$$

The results of the extrapolation $\hat{\mathscr{U}}$ and the analytical results are given at Figures 3.5 and 3.6. The numerical solver uses $R_{\max} = 10^2$, $K = 3$, $M = 5$ and $\beta = 0.1 \cdot 2^t$ for $t = -2 : 0.5 : 5$. The solver is applied to 2 grids of $N_r \times N_\theta = (128 \times 128)$ as the coarse grid (with $\Delta_r \approx 0.037$), and $N_r \times N_\theta = (256 \times 256)$ as the fine grid (with $\Delta_r \approx 0.018$), in order to use Richardson extrapolation for $\hat{\mathscr{U}}$.

3.2.3 Electrophoresis for large β

When the electric field is large, nonlinear effects dominate the solution. In order to investigate the solution for strong electric fields, we have run the solver on the TAMNUN HPC cluster [28]. Large grid sizes are used to verify that the numerical solution converges when the grid becomes finer and that no numerical artifacts are present in the solution. Moreover, several $R_{max} \in \{10, 30, 100\}$ values are used, to make sure it has no major impact on the solution. The results are shown in the following figures. The steady-state velocity results are shown for several grid resolutions and several R_{max} values in Figure 3.7, and the differences between the numerical results for steady-state velocity and the linear regime are shown in Figure 3.8. The estimated rate of convergence is shown in Figure 3.9.

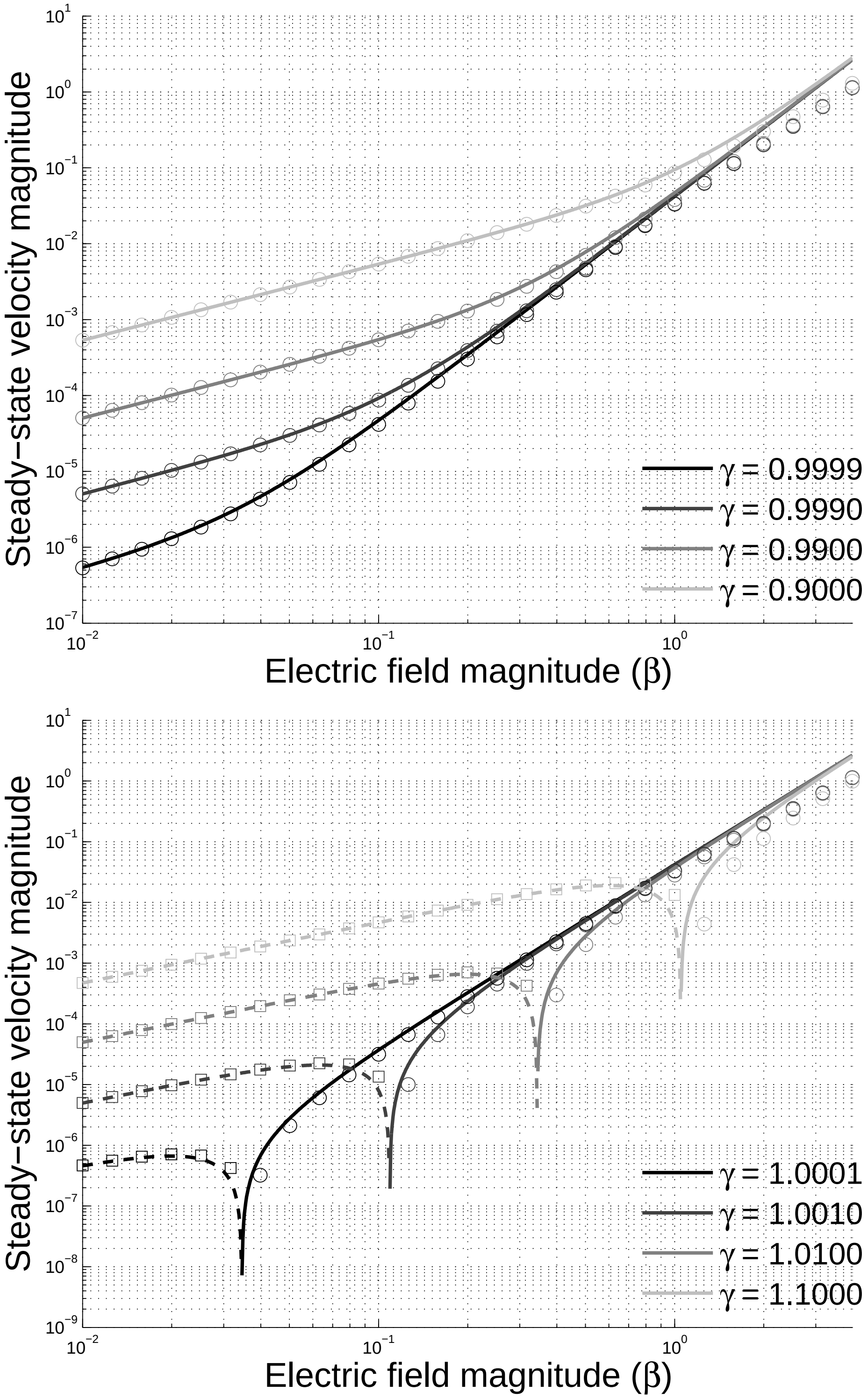

Figure 3.1: Steady-state velocity $\mathscr{U}$ as a function of the electric field magnitude β, for $\gamma < 1$ (top) and $\gamma > 1$ (bottom). The analytic cubic solution (3.11) is represented by curves (solid for $\mathscr{U} > 0$, dashed for $\mathscr{U} < 0$), and the numerical results are represented by symbols (circles for $\mathscr{U} > 0$, squares for $\mathscr{U} < 0$). Note that linear and cubic regimes match well the numerical results, including the critical β behaviour (3.13).

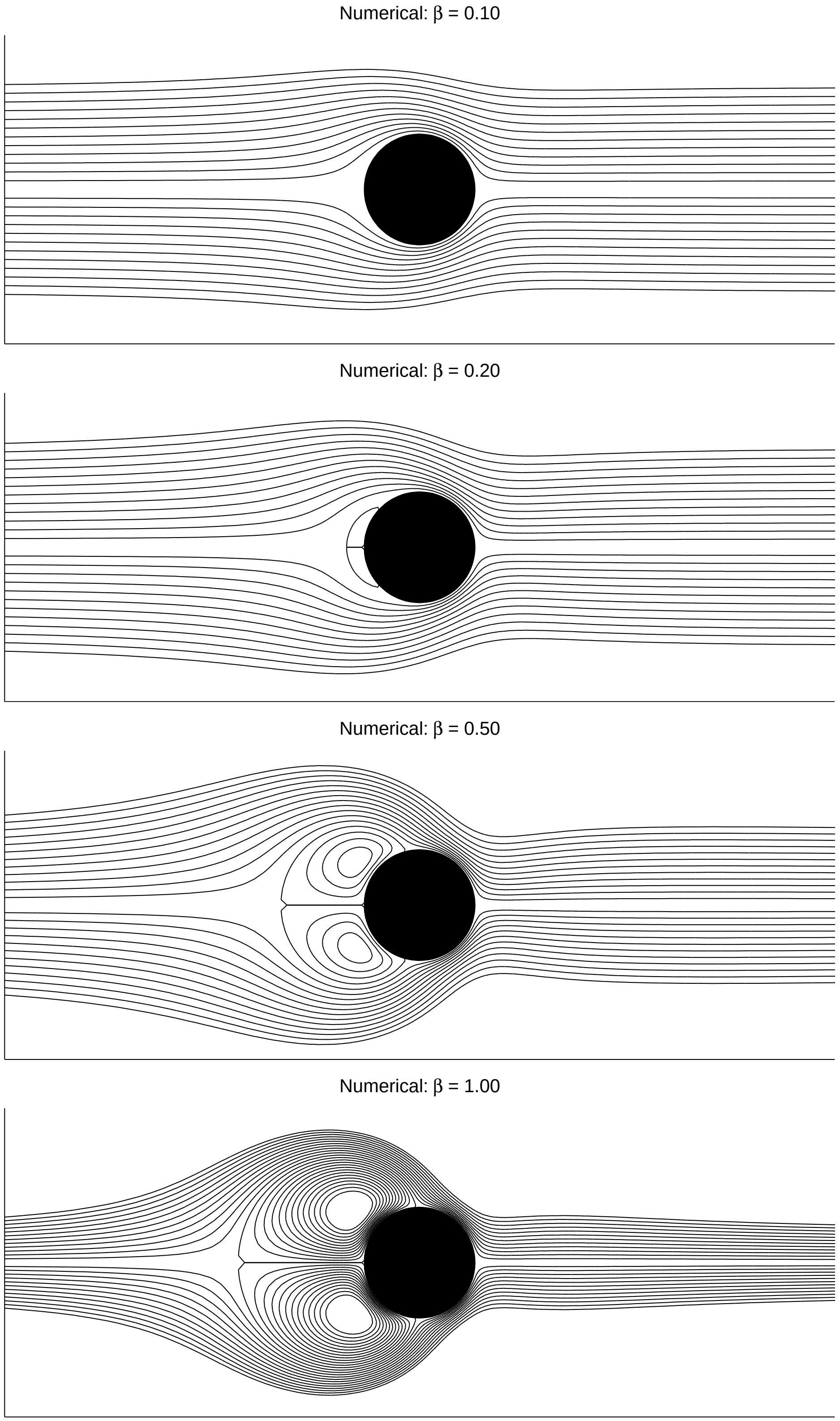

Figure 3.2: Streamlines of the flow for various electric field β values: numerical results for $\gamma = 0.9$. The steady-state velocity $\mathscr{U}$ is positive, so the ion-exchanger drifts to the right.

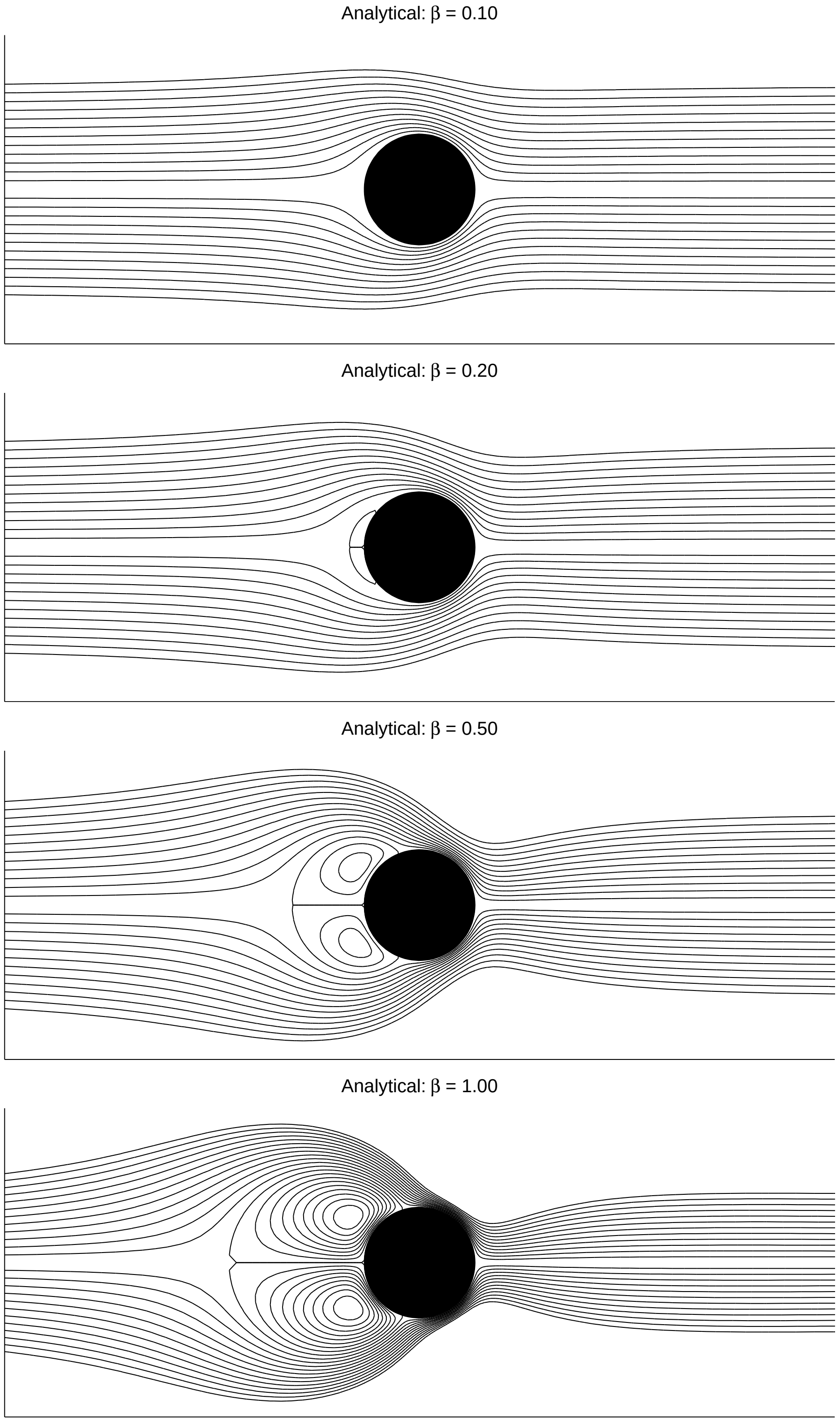

Figure 3.3: Streamlines of the flow for various electric field β values: analytical approximation for $\gamma = 0.9$ (see Section 3.1). The steady-state velocity $\mathscr{U}$ is positive, so the ion-exchanger drifts to the right.

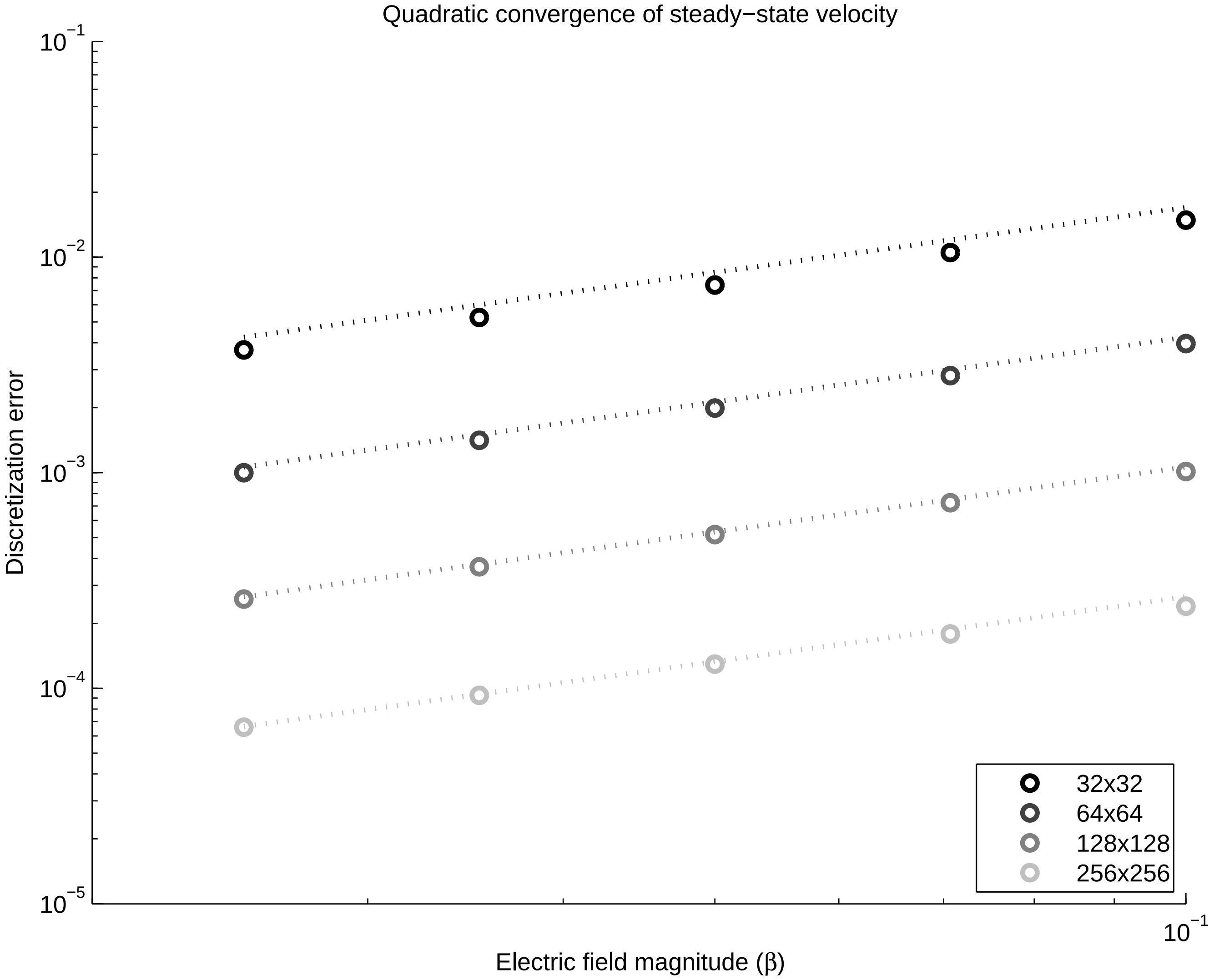

Figure 3.4: The discretization error $\hat{\mathscr{U}}^h - \mathscr{U}_1$ between the numerical results and the linear asymptotic solutions for $\alpha = 0$, Du $= 0.5$, $\bar{\zeta} = 6$ and $R_{max} = 100$. The symbols represent the discretization errors for various grid sizes, and the dotted lines correspond to linear functions of β, scaled by 4^n. The graph shows that, as the number of grid points in each dimension increases by a factor of 2, the discretization error decreases by a factor of 4.

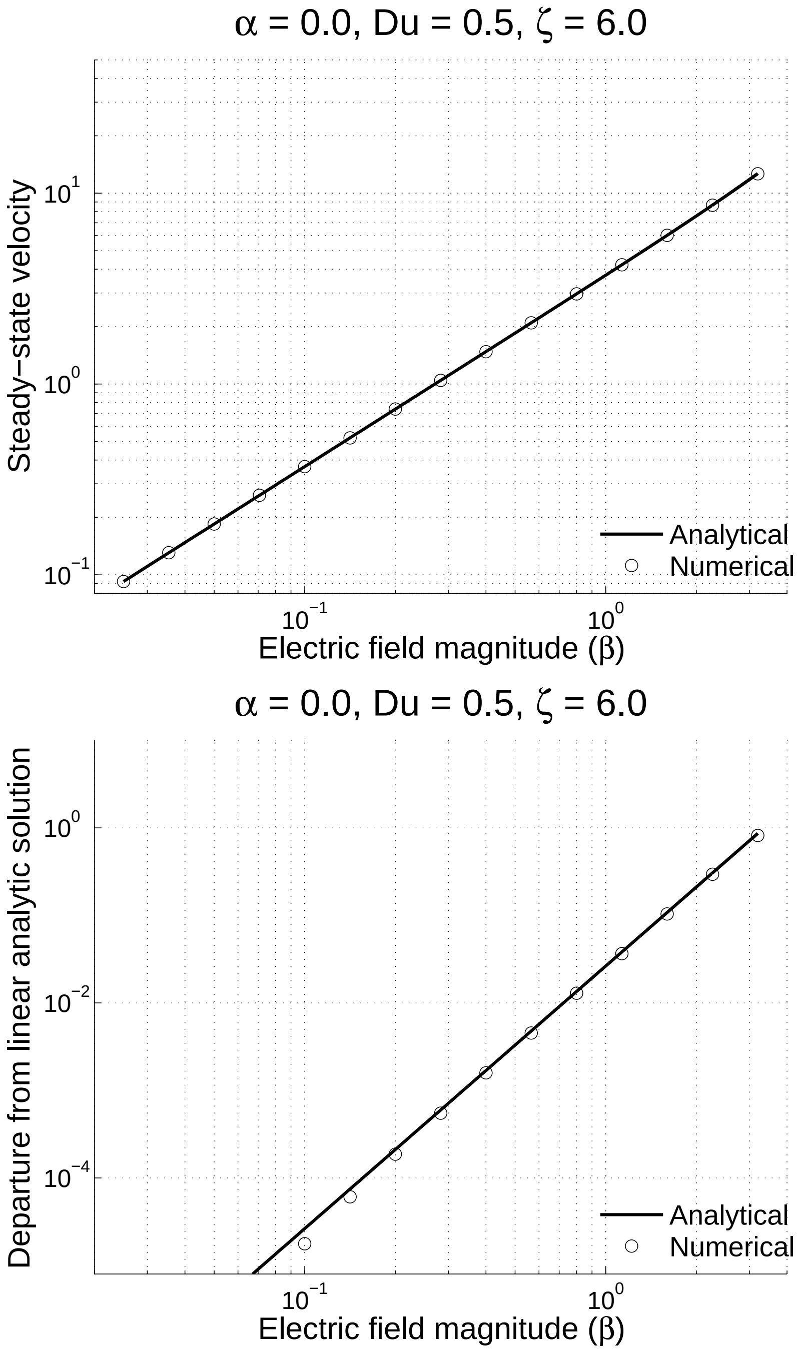

Figure 3.5: Steady-state velocity $\mathscr{U}$ as a function of the electric field magnitude β, for highly-charged surface particle electrophoresis (with $\alpha = 0, \mathrm{Du} = 0.5, \bar{\zeta} = 6$). The analytic solution velocity is represented by solid lines, and the numerical results are represented by symbols. The top figure shows the full numerical solution compared to the analytical one (3.11), and the bottom figure compares the analytic cubic correction [27] to the difference between the numerical results and the linear regime.

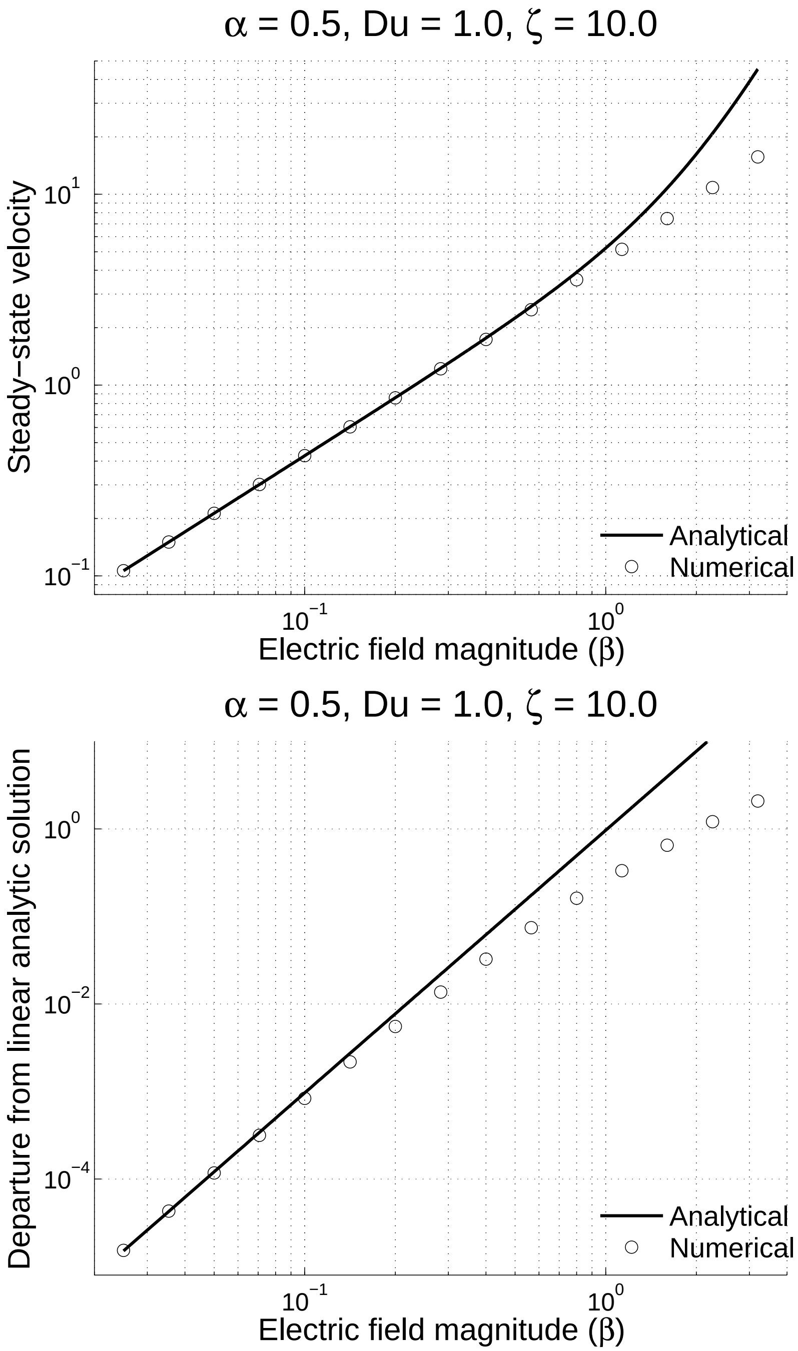

Figure 3.6: Steady-state velocity $\mathscr{U}$ as a function of the electric field magnitude β, for highly-charged surface particle electrophoresis (with $\alpha = 0.5, \mathrm{Du} = 1, \bar{\zeta} = 10$). The analytic solution velocity is represented by solid lines, and the numerical results are represented by symbols. The top figure shows the full numerical solution compared to the analytical one (3.11), and the bottom figure compares the analytic cubic correction [27] to the difference between the numerical results and the linear regime. When β is large, the analytical cubic solution loses its validity because higher-order terms become significant.

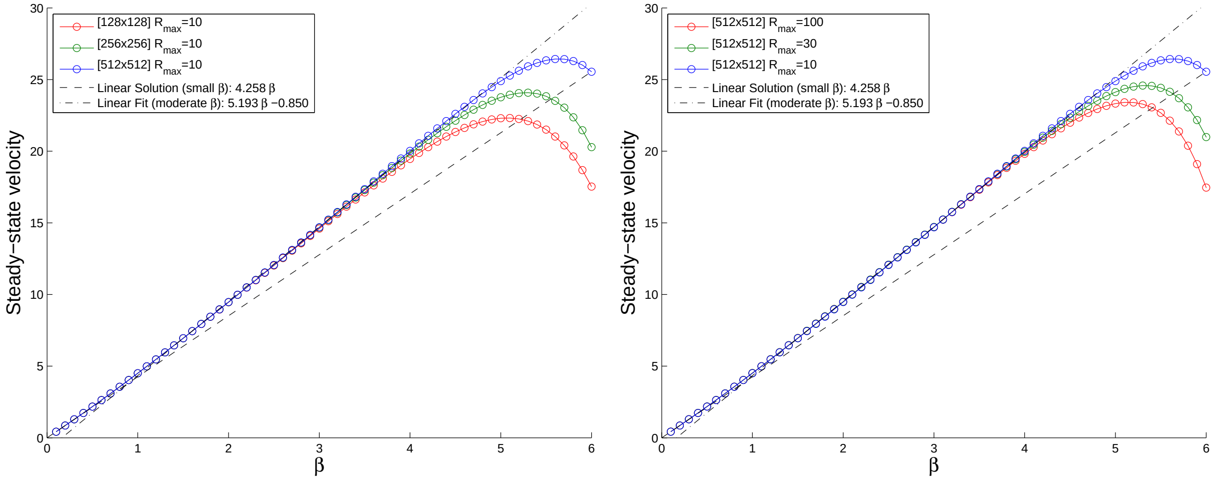

Figure 3.7: Large β results for $\mathscr{U}(\beta)$ ($\mathrm{Du} = 1$, $\bar{\zeta} = 10$, $\alpha = 0.5$). The left figure shows the numerical results of $\mathscr{U}(\beta)$ for $R_{max} = 10$ and the following grid sizes: 128×128, 256×256, 512×512. The right figure shows the numerical results of $\mathscr{U}(\beta)$ for 512×512 grid for $R_{max} = 100, 30, 10$. For small values of β, the linear regime of $\mathscr{U}(\beta)$ is dominant. For moderate β values, the steady state velocity is very close to an affine function of β. However, for larger values of β, the numerical solution deviates significantly from the linear regime, due to convergence problems at $\beta > 6$, and for $\beta \sim 7$, the solver fails to converge. The numerical convergence depends on the grid resolution near the particle surface – as it becomes finer (when N_r increases or R_{max} decreases), the solver convergence improves.

The numerical solver converges well for $\beta < 5$ and the resulting steady-state velocity for moderate β values is very close to being an affine function of β. When β becomes large, the salt concentration C near the front of the particle (at $\theta \approx 0$) approaches zero, while creating a wake behind the particle (at $\theta \approx \pi$) as shown in Figures 3.10, 3.11 and 3.12. This results in high gradients in $\log C$ near the front of the particle and, therefore, a strong electric field $\boldsymbol{E}$ forms near the particle surface (Figures 3.13, 3.14 and 3.15) due to effective boundary conditions in (1.17). From Figs. 3.7-3.8 we conclude that even our most accurate numerical solution is valid only up to $\beta = 5$ approximately, and therefore we only further examine solutions in this range.

The streamline pattern does not change drastically for different electric field magnitudes. The resulting fluid flow is nearly proportional to that of the Stokes creeping flow, as shown in Figure 3.16 and 3.17. The difference between $\Psi(r, \theta)$ and $\Psi(r, \pi - \theta)$ is shown in Figure 3.18, demonstrating that the flow is almost symmetric (with relative difference up to approximately 0.5%) for $\beta < 5$.

The numerical results show that a boundary layer starts to form near the front of the particle for strong electric fields (see Figures 3.19 and 3.20). The width of the boundary layer with respect to the electric field scales, to a good approximation, as: $\bar{\rho}(\beta) \approx 0.2e^{-0.5\beta}$. This causes numerical problems for the current solver at large β, due to insufficient grid resolution at the boundary layer of C and $\boldsymbol{E}$.

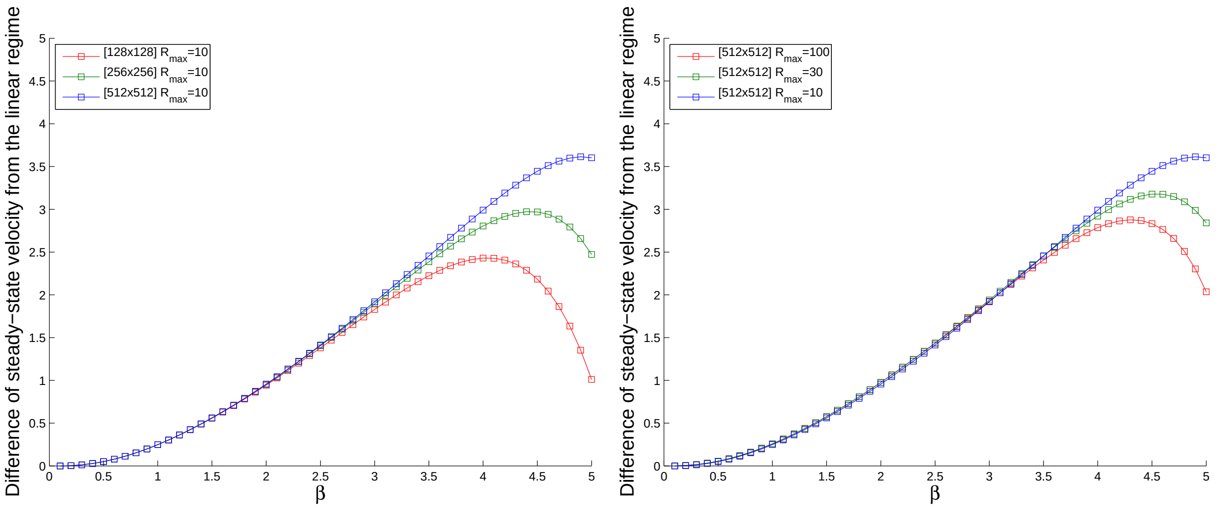

Figure 3.8: Large β results for $\mathscr{U}(\beta) - \beta\mathscr{U}_1$ (Du $= 1$, $\bar{\zeta} = 10$, $\alpha = 0.5$ and $\mathscr{U}_1 \approx 4.258$ is the linear regime coefficient). The left figure shows the numerical results of $\mathscr{U}(\beta)$ for $R_{max} = 10$ and the following grid sizes: 128×128, 256×256, 512×512. The right figure shows the numerical results of $\mathscr{U}(\beta)$ for 512×512 grid for $R_{max} = 100, 30, 10$. It is seen that the difference is indeed nearly an affine function of β in the large-β regime, so long as the numerical solution remains sufficiently accurate.

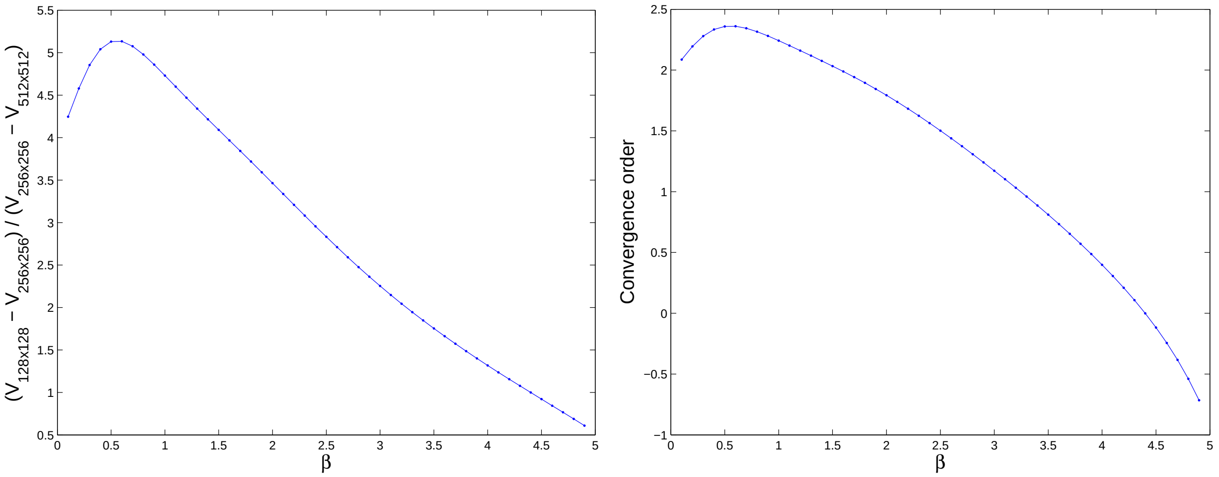

Figure 3.9: Rate of convergence estimation for $\mathscr{U}$ at large β (Du $= 1$, $\bar{\zeta} = 10$, $\alpha = 0.5$), based on the numerical results of $\mathscr{U}(\beta)$, for the grid sizes of 128×128, 256×256, 512×512. The rate of convergence $r(\beta)$ is estimated by $r(\beta) \triangleq \left(\mathscr{U}^{4h}(\beta) - \mathscr{U}^{2h}(\beta)\right) / \left(\mathscr{U}^{2h}(\beta) - \mathscr{U}^{h}(\beta)\right)$ (left). The order of convergence is estimated by $p(\beta) \triangleq \log_2 r(\beta)$ (right). For small β, $p(\beta) \to 2$, showing quadratic convergence for the linear regime. For moderate β, the convergence rate slows down, and it deteriorates for large β values.

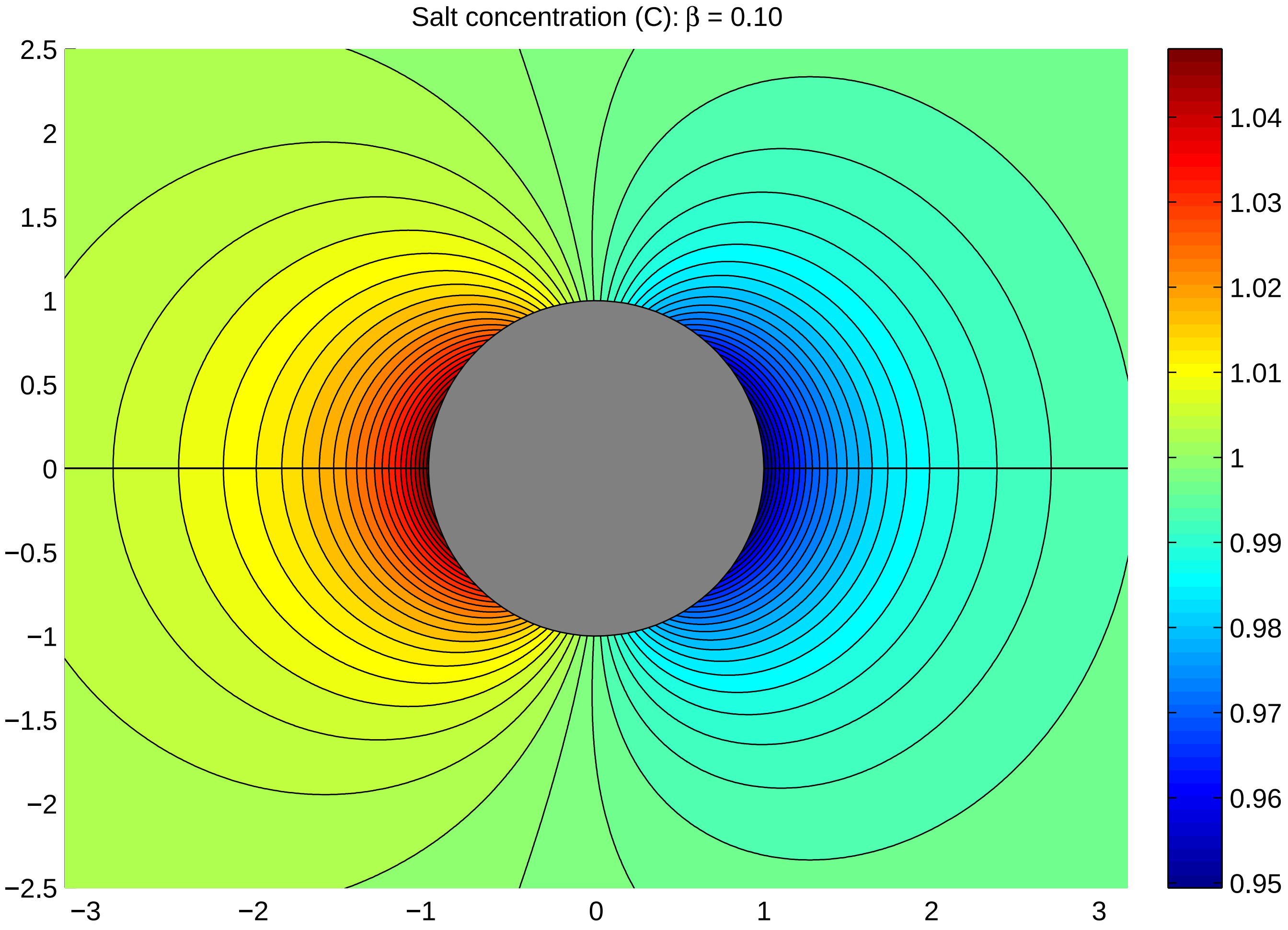

Figure 3.10: $\beta = 0.1$ results for C: grid size 512×512, $R_{max} = 10$, $\mathrm{Du} = 1$, $\bar{\zeta} = 10$, $\alpha = 0.5$. Here the linear regime dominates the solution for $C_\beta(r,\theta) \approx 1 - \beta\frac{\cos\theta}{2r^2}$.

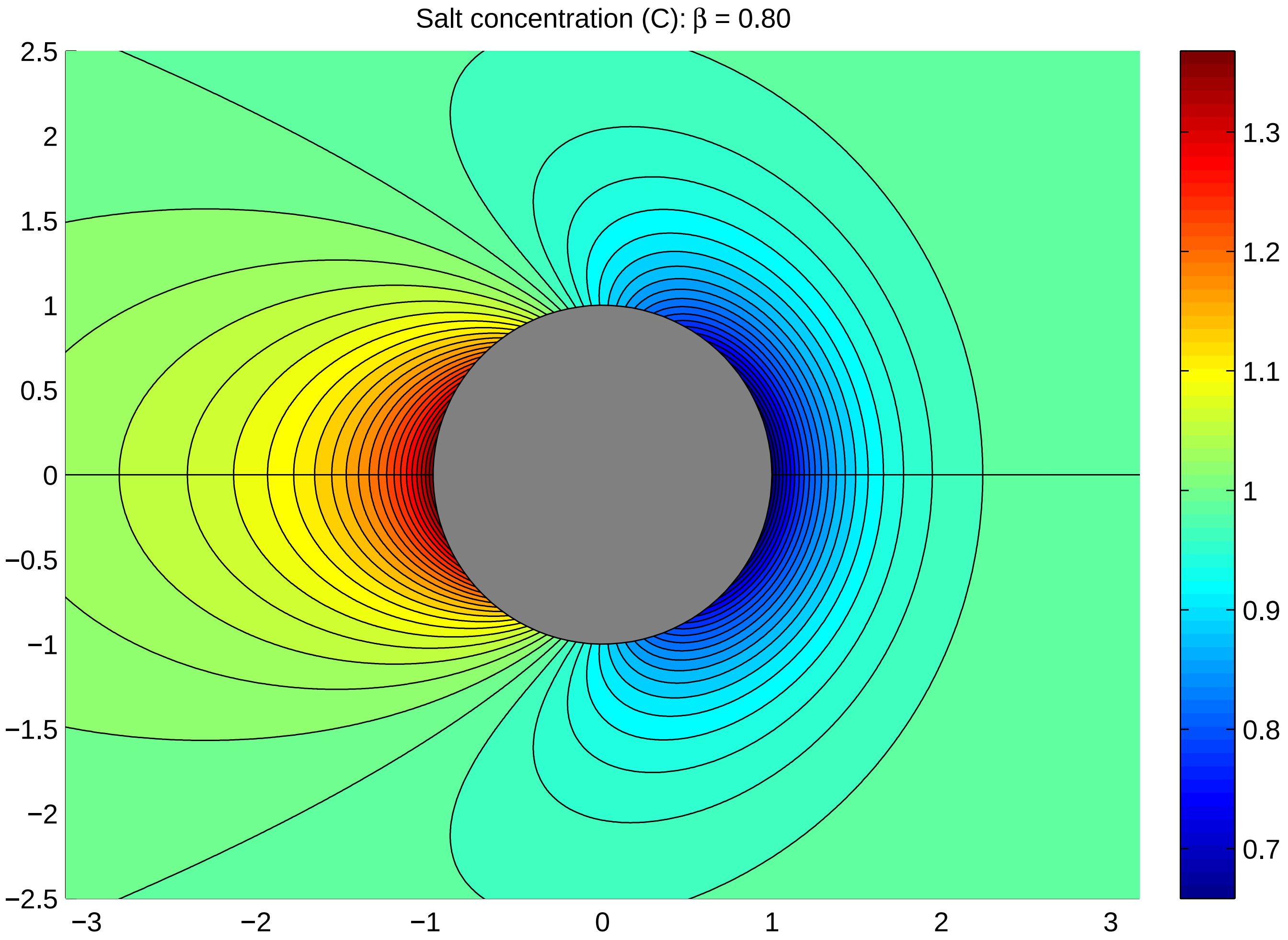

Figure 3.11: $\beta = 0.8$ results for C: grid size 512×512, $R_{max} = 10$, Du $= 1$, $\bar{\zeta} = 10$, $\alpha = 0.5$. For moderate β values, the salt concentration starts to form a wake behind the particle ($\theta \approx \pi$).

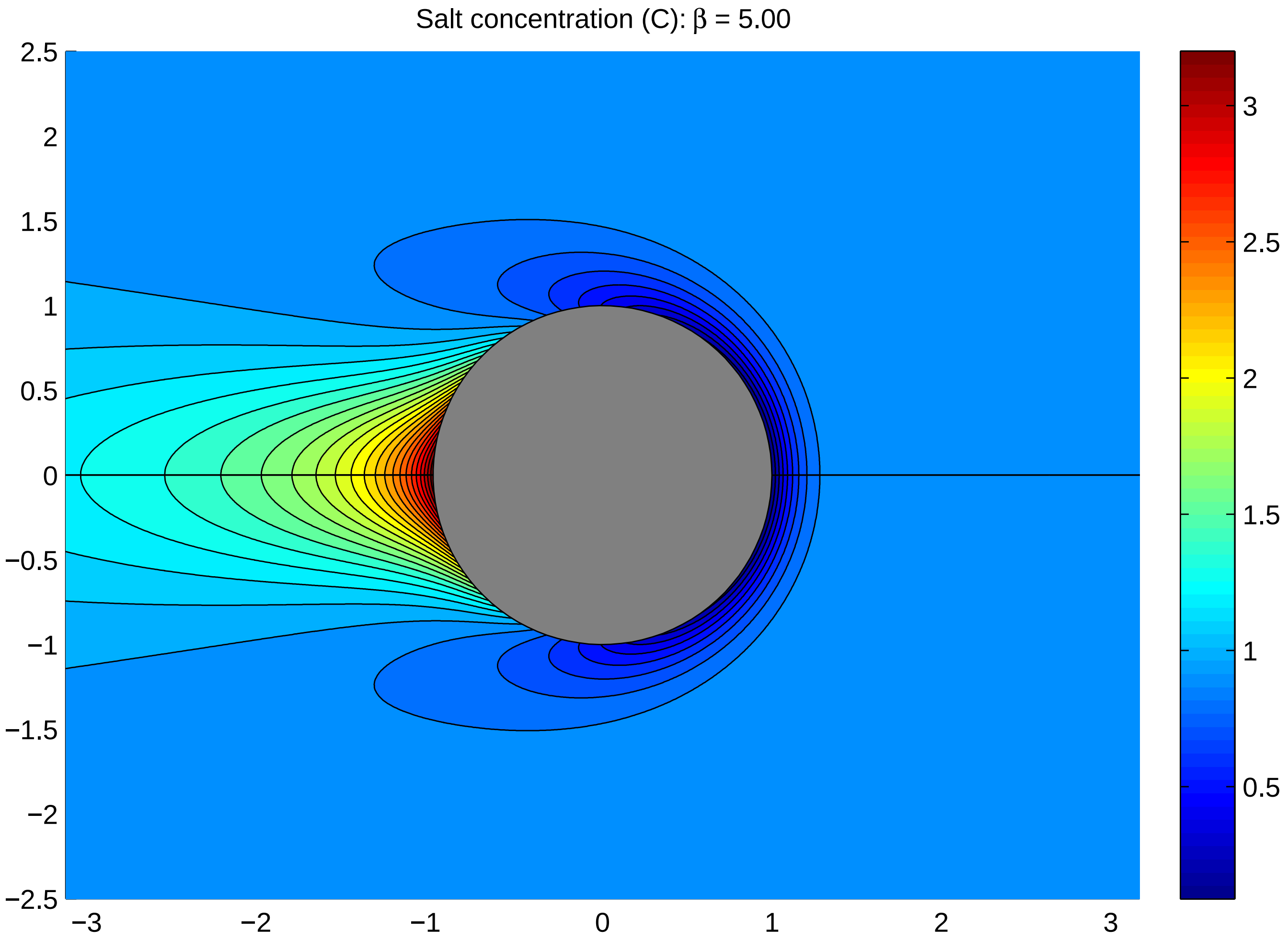

Figure 3.12: $\beta = 5.0$ results for C: grid size 512×512, $R_{max} = 10$, Du $= 1$, $\bar{\zeta} = 10$, $\alpha = 0.5$. For large values of β, there is a significant drop in salt concentration at the front of the particle – C approaches zero at $\theta \approx 0$, such that $\frac{\partial C}{\partial r}$ becomes large (since the change in C happens on a small length scale). Due to the boundary conditions of the electrophoresis problem (given by (1.17)), the electric field radial component given by $E_r = -\frac{\partial \Phi}{\partial r} = -\frac{1}{C}\frac{\partial C}{\partial r}$, becomes singular where $C \to 0$.

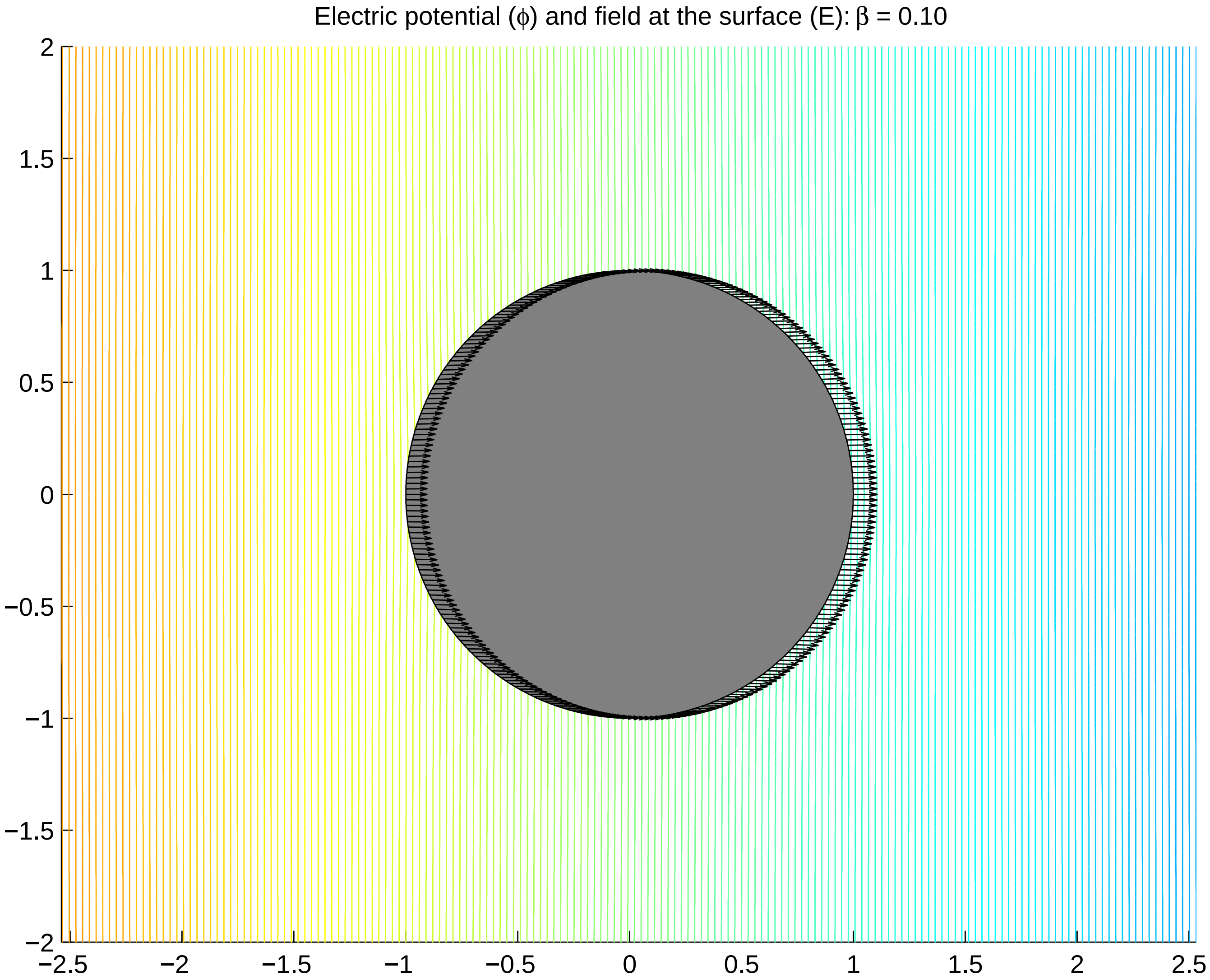

Figure 3.13: $\beta = 0.1$ results for $\varPhi$: grid size 512×512, $R_{max} = 10$, $\mathrm{Du} = 1$, $\bar{\zeta} = 10$, $\alpha = 0.5$. The electric potential and the electric field results are multiplied by β^{-1}, so the linear regime appears unchanged for different values of β. Here the linear regime dominates the solution for $\varPhi_\beta(r,\theta) \approx -\beta r \cos\theta$, hence the electric field is constant for weak electric fields: $\boldsymbol{E} \approx \beta\hat{\boldsymbol{\imath}}$.

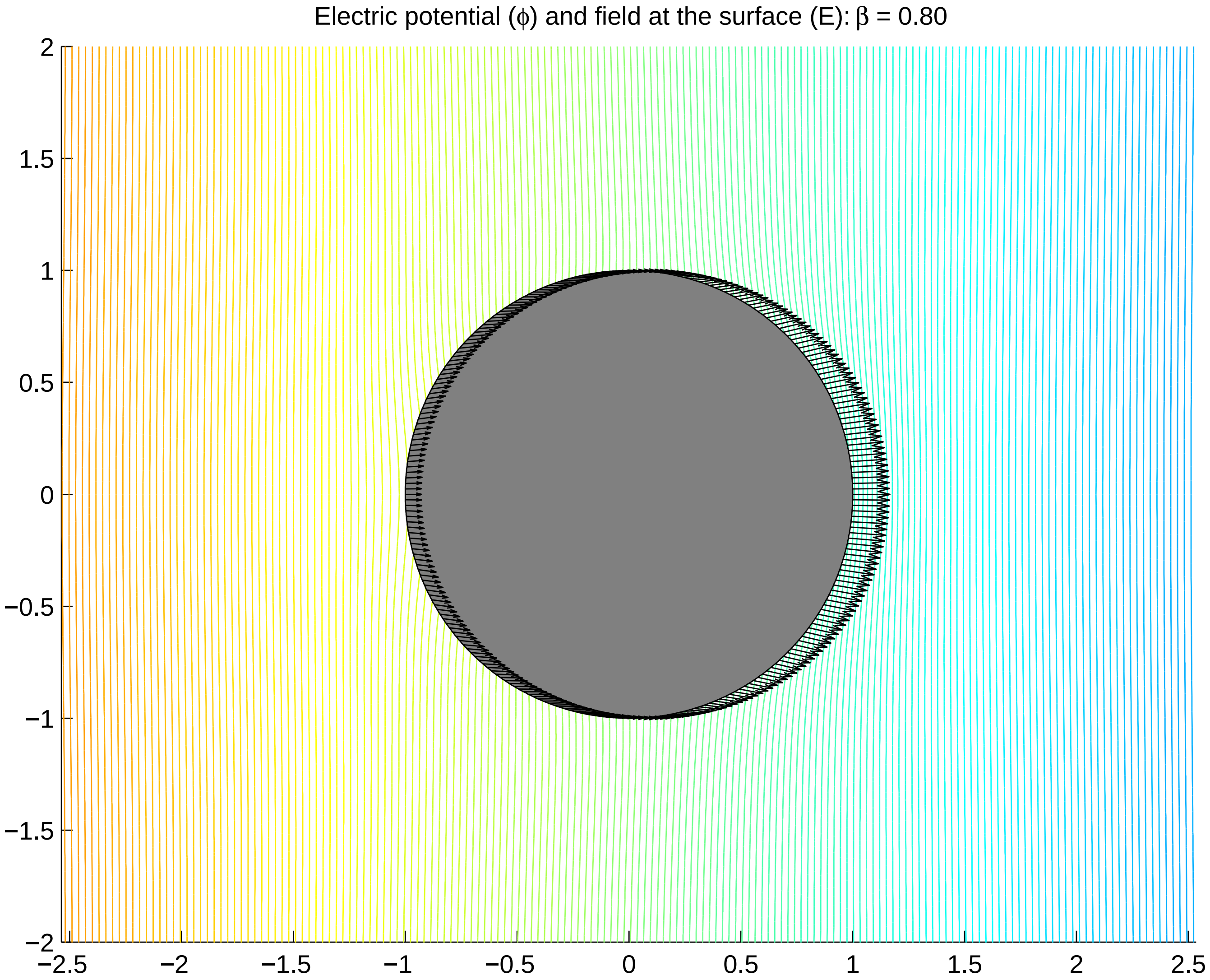

Figure 3.14: $\beta = 0.8$ results for Φ: grid size 512×512, $R_{max} = 10$, $\mathrm{Du} = 1$, $\bar{\zeta} = 10$, $\alpha = 0.5$. The electric potential and the electric field results are multiplied by β^{-1}, so the linear regime appears unchanged for different values of β. For moderate values of β, the symmetry breaks and the electric field becomes stronger near the front of the particle than at the back.

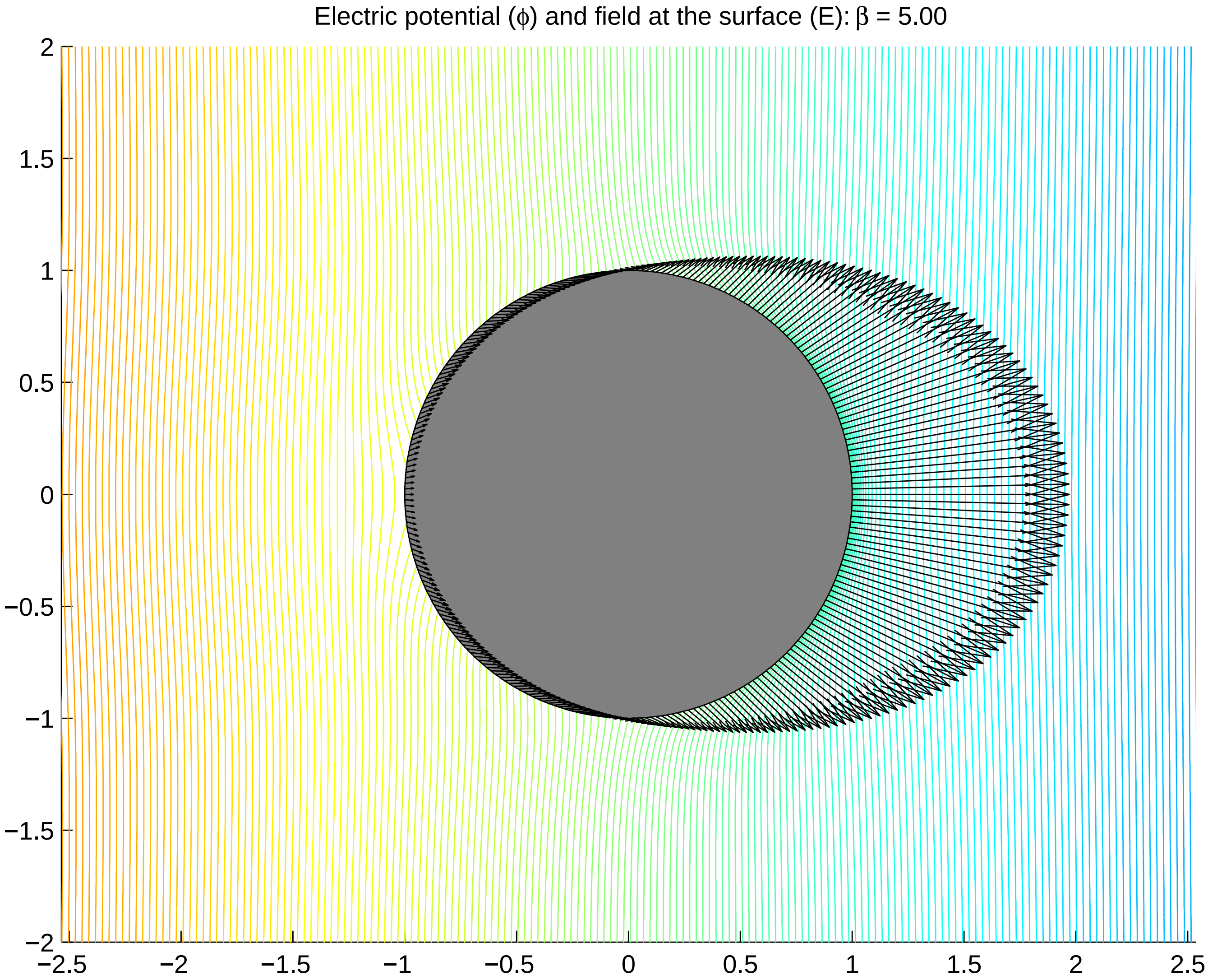

Figure 3.15: $\beta = 5.0$ results for Φ: grid size 512×512, $R_{max} = 10$, $\mathrm{Du} = 1$, $\bar{\zeta} = 10$, $\alpha = 0.5$. The electric potential and the electric field results are multiplied by β^{-1}, so the linear regime appears unchanged for different values of β. For large values of β, the electric field radial component $E_r = -\frac{\partial \Phi}{\partial r}$ at the particle front becomes much larger than the electric field at the particle back.

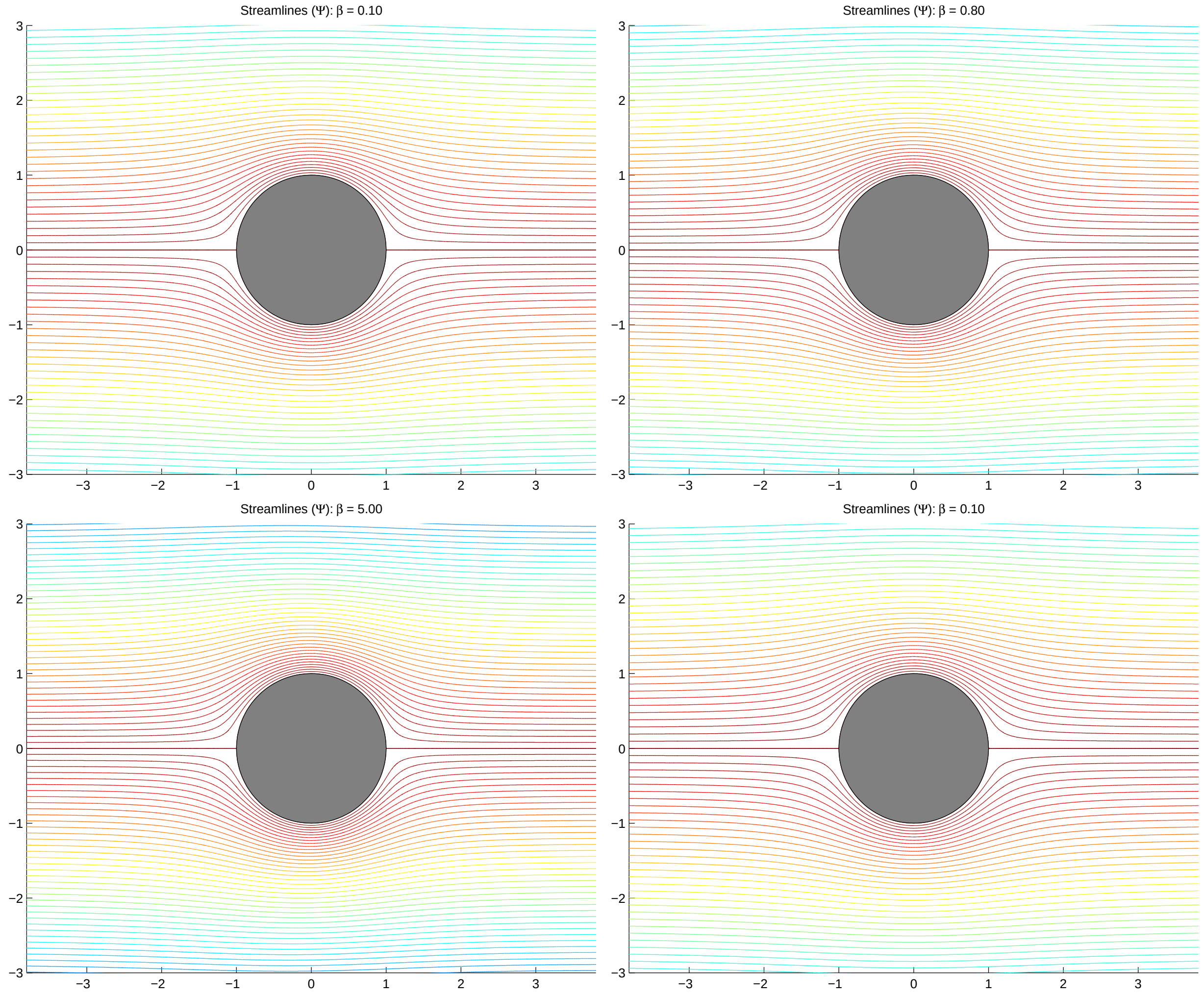

Figure 3.16: $\beta = 0.1, 0.8, 5.0$ and Stokes streamlines results: grid size 512×512, $R_{max} = 30$, Du $= 1$, $\bar{\zeta} = 10$, $\alpha = 0.5$. The numerical results for Ψ are scaled by β^{-1}, so the linear regime will appear the same for different β values. Note that the numerical results show a flow pattern very similar to that of the analytical Stokes flow.

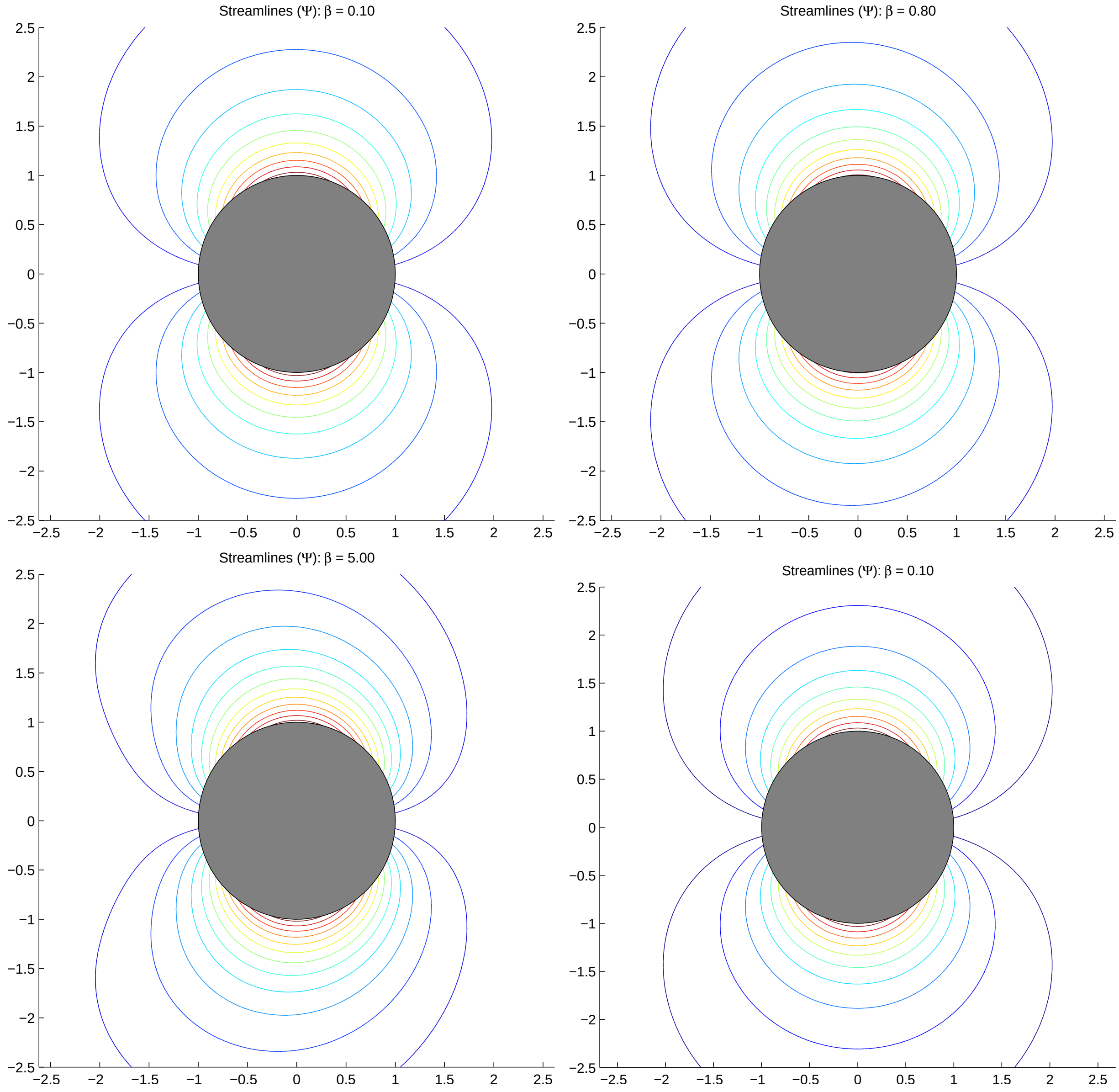

Figure 3.17: $\beta = 0.1, 0.8, 5.0$ and Stokes streamlines results after subtraction of the uniform flow streamfunction ($\frac{\mathscr{U}}{2} r^2 \sin^2\theta$): grid size 512×512, $R_{max} = 30$, $\mathrm{Du} = 1$, $\bar{\zeta} = 10$, $\alpha = 0.5$. The numerical results for Ψ are scaled by β^{-1}, so the linear regime will appear the same for different β values. Even for the largest beta we could compute, the velocity remains fairly symmetric and "uneventful" also in the framework of the "laboratory" rather than that of the particle.

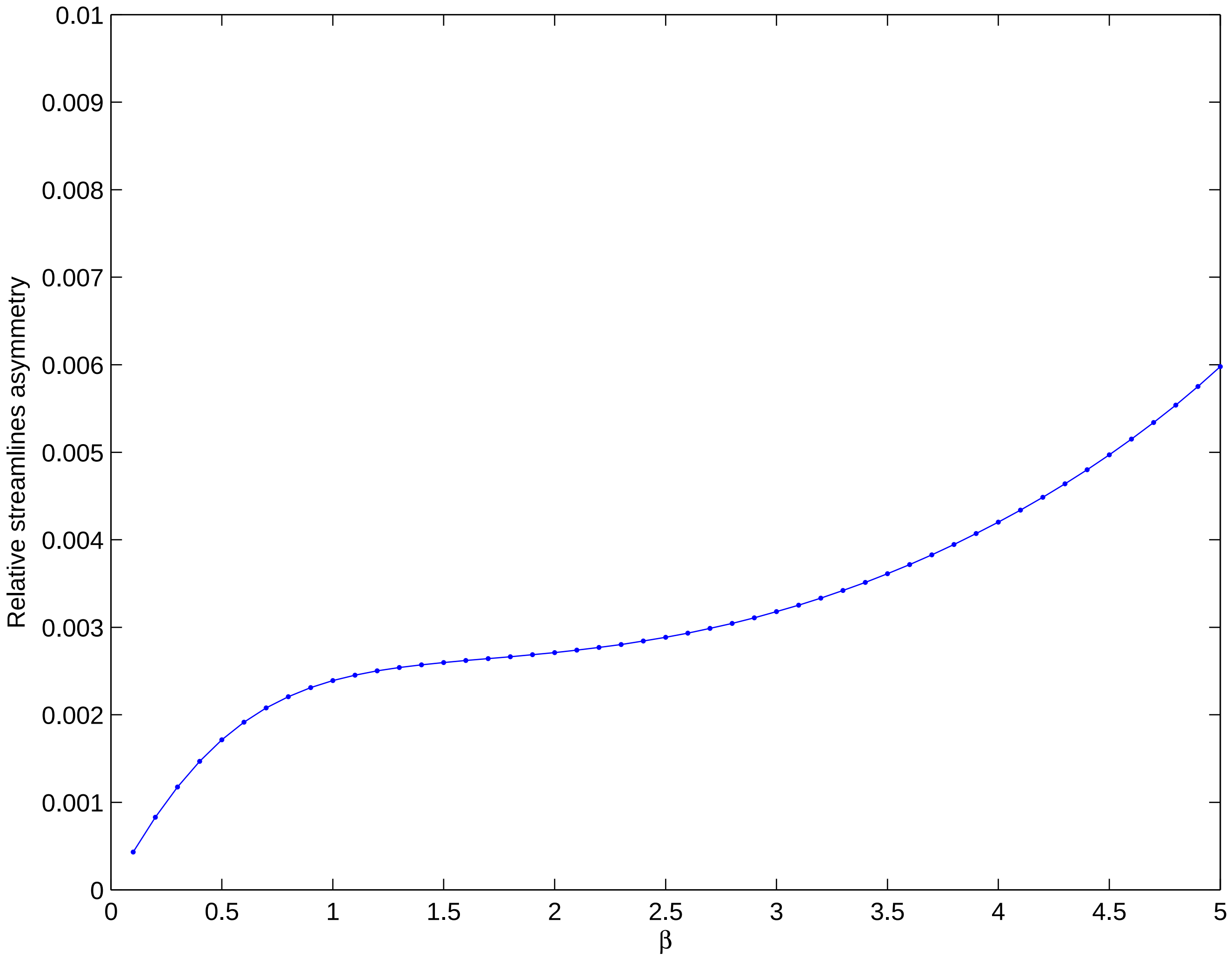

Figure 3.18: Streamline results: grid size 512 × 512, $R_{max} = 10$, Du $= 1$, $\bar{\zeta} = 10$, $\alpha = 0.5$. The relative streamline asymmetry is computed by $\|\Psi(r,\theta) - \Psi(r,\pi-\theta)\|/\|\Psi(r,\theta)\|$ as a function of β. The results show that the streamline pattern is almost symmetric, even for moderately large β values.

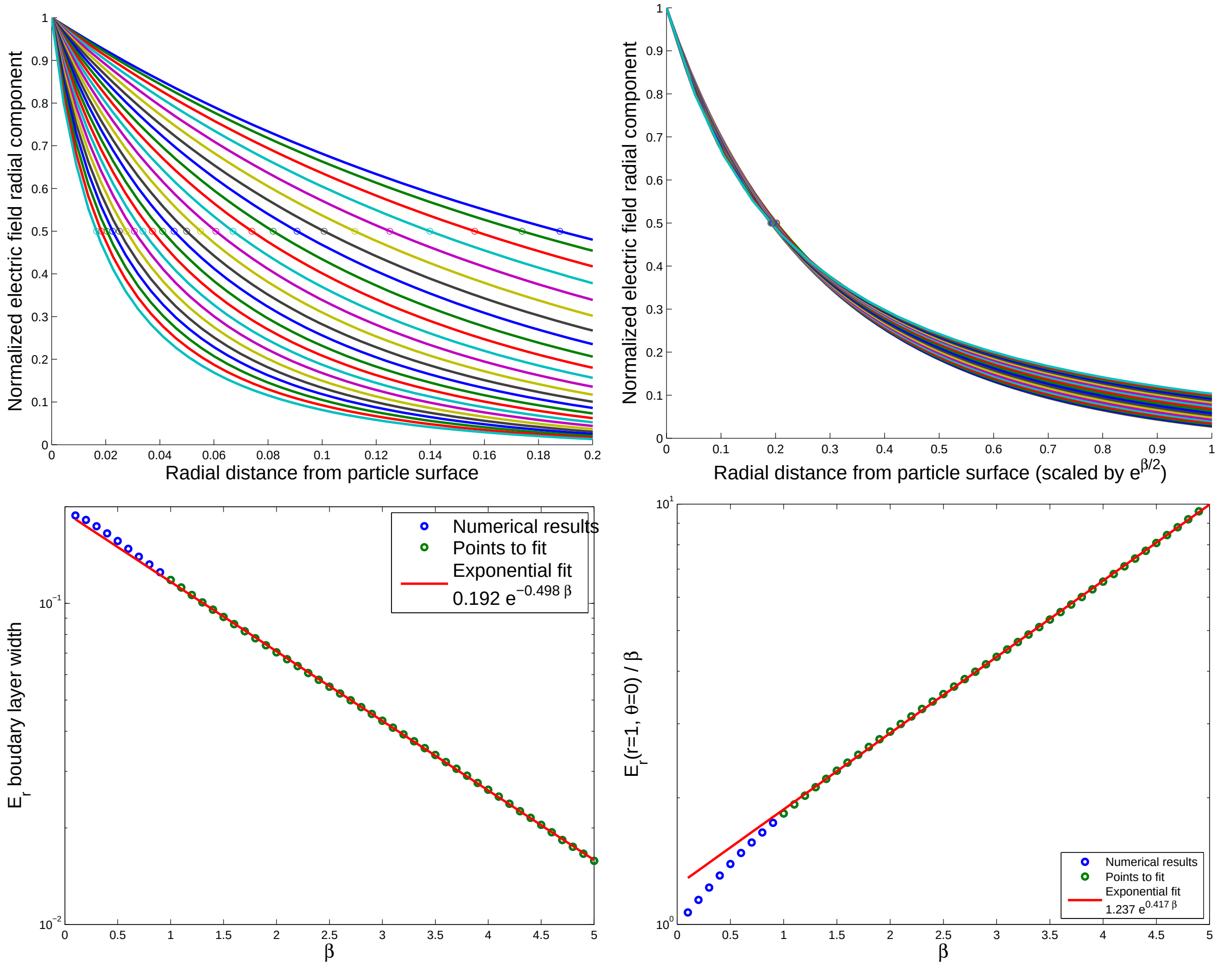

Figure 3.19: Electric field boundary layer formation: grid size 512×512, $R_{max} = 30$, $\mathrm{Du} = 1$, $\bar{\zeta} = 10$, $\alpha = 0.5$. The upper left graph shows normalized electric field radial component E_r near the particle front ($\theta = 0$). The numerical results for $e(\rho) = E_r(r = 1 + \rho, \theta = 0)$ are normalized by $\hat{e}(\rho) = (e(\rho) - \beta) \, / \, (e(0) - \beta)$, where $e(\rho = \infty) = \beta$. The normalized electric field satisfies $\hat{e}(\rho) \in [0, 1]$ and is shown as function of $\rho = r - 1$ at the upper left graph and as function of $\rho e^{0.5\beta}$ at the upper right graph. In order to estimate the width of the boundary layer, for each β, we compute the value of $\bar{\rho}(\beta) = \min_{\rho}\{\rho : \hat{e}(\rho) < 0.5\}$. Each such $\bar{\rho}(\beta)$ is plotted at the left graph as a solid dot on the corresponding line. In addition, $\bar{\rho}(\beta)$ is plotted as a function of β using logarithmic y axis at the lower left graph to show that $\bar{\rho}(\beta) \approx 0.2e^{-0.5\beta}$. Moreover, $E_r(\beta; r = 1, \theta = 0)/\beta$ grows exponentially as a function of β, as shown at the lower right graph. Thus, a boundary layer is indeed forming near the front of the particle.

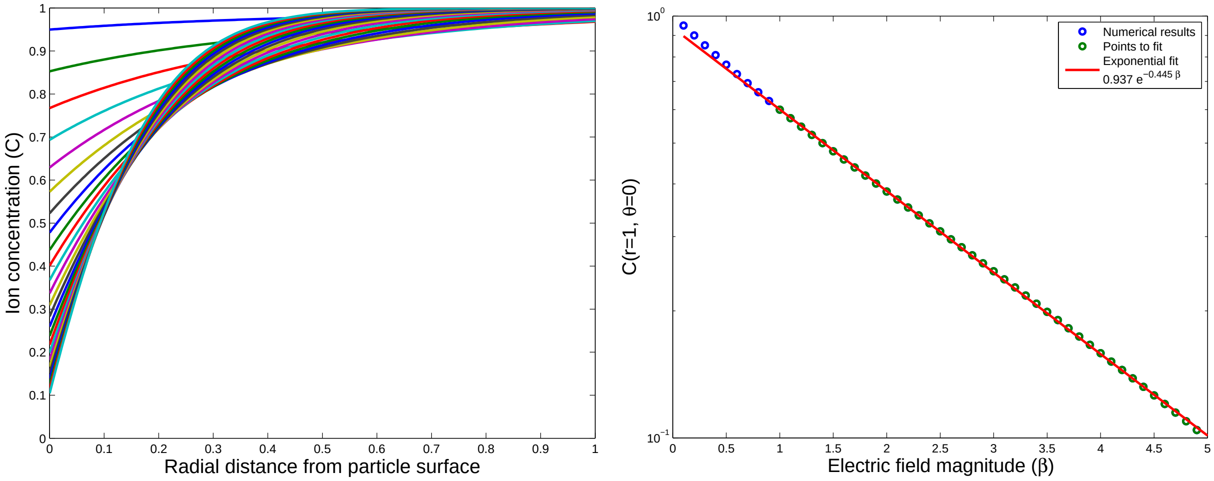

Figure 3.20: Ionic concentration boundary layer formation: grid size 512 × 512, $R_{max} = 30$, Du $= 1$, $\bar{\zeta} = 10$, $\alpha = 0.5$. The left graph shows the ionic concentration C near the particle front ($\theta = 0$). In addition, $C_\beta(r = 1, \theta = 0)$ is plotted as a function of β using logarithmic y axis at the right graph to show that $C_\beta(r = 1, \theta = 0) \approx 0.94e^{-0.45\beta}$. The ionic concentration becomes very close to zero for $\beta \approx 7$ at $r = 1$, creating a singularity near the particle front.

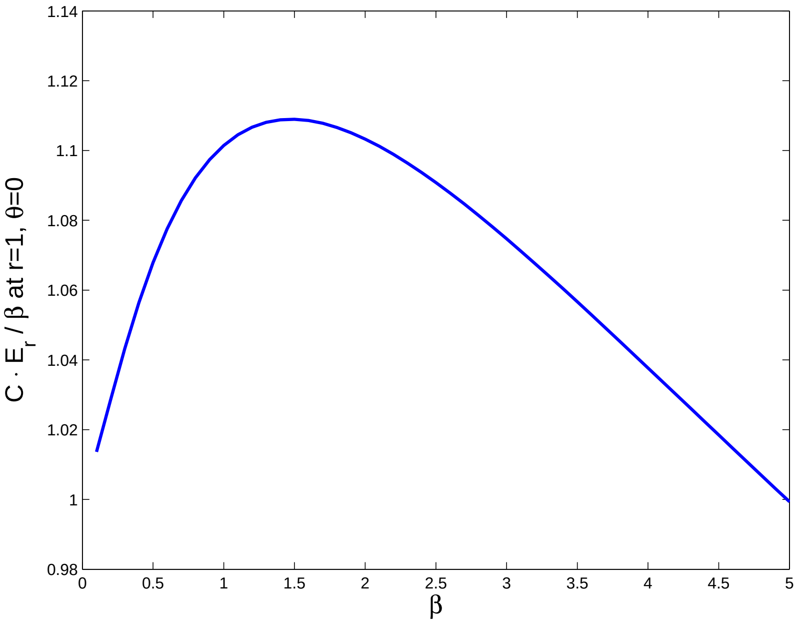

Figure 3.21: The product of $C(r = 1, \theta = 0)E_r(r = 1, \theta = 0)/\beta$ as a function of β.

3.3 Discussion

A numerical framework for a nonlinear electrokinetic transport system is developed. The framework is used to construct numerical solvers for two electrokinetic problems: the migration of ion-exchangers and surface-charged inert particle electrophoresis. The numerical results for weak electric field exhibit a good correspondence to the asymptotic models as shown above. Two interesting phenomena (cubic regime and downstream vortex) were predicted for the ion-exchanger using the numerical results, and were verified using an asymptotic expansion of the nonlinear system. The electrophoresis problem was solved on a high-resolution grid to confirm the asymptotic cubic correction of the steady-state velocity, as well as to provide insights into the solution behavior for moderately strong electric fields. For strong electric field, the nonlinear effects become dominant and a boundary-layer structure begins to form. Due to diminishing ion concentration C at the front of the particle, a boundary layer of thickness approximately $0.2e^{-0.5\beta}$ is formed with respect to the electric field $\boldsymbol{E}$ at the front of the particle. The value of E_r/β at $(r = 1, \theta = 0)$ is given approximately by $e^{\beta/2}$, while the value of $C(r = 1, \theta = 0)$ is given approximately by $e^{-\beta/2}$, and their product remains close to 1 for all β that we could compute (as shown in Figure 3.21). However, the velocity of the particle remains nearly linear with β and the velocity field is very close to being symmetric. Finally, we emphasize the importance of the analytical asymptotic derivation. This approach provides effective boundary conditions for the macroscale problem and thus eliminates the inherent physical scale disparity. This elimination enables the construction of the numerical scheme. Furthermore, the analytical asymptotic solutions for the weak field regime enable important mutual verification of the numerical results.

3.4 Future Work

Future work may include using the numerical results described in this work to develop an analytical solution for the strong field regime of the electrophoresis problem, which is of interest. The solver may be used to investigate how the solution (and the evolving boundary layer, in particular) depends on the problem parameters. Optimizing the direct sparse linear solver can be done by using the ordering data from previous calls, or replacing the full Newton step by an iterative linear solver (e.g., using preconditioned Krylov subspace methods [29] or Multigrid), allowing finer grids to be used. In order to achieve high numerical accuracy, the grid discretization should be made finer near the particle surface (at $r = 1$), while R_{max} can be made smaller. Moreover, a different discretization scheme may be used for the r coordinate, to have higher grid resolution near the particles. In addition, the continuation of the solution between different β values should advance in finer steps, to ensure Newton method convergence. The discretization of boundary conditions far away from the particle may be improved by including the linear correction from the asymptotic solution, or reformulation of far-away boundary condition according to the asymptotic behavior of the analytic solution far away from the particle. Moreover, the framework may be used for automatic construction of Newton solvers for other nonlinear PDE systems, for example, for particles with different surface properties, or different particle shape.

Appendix A

Effective Boundary Conditions

A.1 Debye layer scaling

We define the scaled coordinate ρ (normal to ion-exchanger surface), where $\delta \ll 1$:

$$\rho = \frac{r-1}{\delta}. \tag{A.1}$$

Thus, the electric potential and the ionic concentrations are $O(1)$ at the Debye layer ($r \approx 1$):

$$\varphi(\rho,\theta) \sim \varphi, \tag{A.2}$$

$$c_\pm(\rho,\theta) \sim c_\pm, \tag{A.3}$$

$$c(\rho,\theta) \sim c, \tag{A.4}$$

$$q(\rho,\theta) \sim q. \tag{A.5}$$

The radial fluxes at the Debye layer are $O(\delta^{-1})$, due to coordinate rescaling:

$$j_\pm(\rho,\theta) \sim \delta^{-1} j_\pm. \tag{A.6}$$

The pressure is $O(\delta^{-2})$, due to momentum balance equations:

$$p(\rho,\theta) \sim \delta^{-2} p. \tag{A.7}$$

The tangential velocity component is $O(1)$ and the radial component is $O(\delta)$.

A.2 Ion Fluxes

In order to match bulk $O(1)$ outer radial ionic flux, the $O(\delta^{-1})$ term of the inner ionic radial flux, $j_{\pm}(\rho,\theta)$ must vanish:

$$\begin{aligned}
-\frac{\partial c_{\pm}}{\partial \rho} \mp c_{\pm}\frac{\partial \varphi}{\partial \rho} &= 0, \\
(\Phi - \varphi) \pm (\log C_{\pm} - \log c_{\pm}) &= 0, \\
\Phi - \varphi \pm \log \frac{C_{\pm}}{c_{\pm}} &= 0.
\end{aligned} \tag{A.8}$$

This relation can be rewritten as a Boltzmann distribution of $c_{\pm}$, where $\lim_{\rho\to\infty} c_{\pm} = C$,

$$c_{\pm} = C \exp\left[\pm(\Phi - \varphi)\right]. \tag{A.9}$$

A.3 Electric Potential

The leading-order Laplace equation in the Debye layer reads

$$\frac{\partial^2 \varphi}{\partial \rho^2} = -q = -\frac{c_+ - c_-}{2} = -\frac{C}{2}\left(\exp\left[+(\Phi - \varphi)\right] - \exp\left[-(\Phi - \varphi)\right]\right) = C\sinh(\varphi - \Phi). \tag{A.10}$$

Denote $\psi = \varphi - \Phi$, so that $\psi(0) = \mathscr{V} - \Phi = \zeta$ and $\psi(\rho \to \infty) = 0$:

$$\begin{aligned}
\frac{\partial^2 \psi}{\partial \rho^2} &= C\sinh(\psi) = 2C\sinh\frac{\psi}{2}\cosh\frac{\psi}{2}, \\
2\frac{\partial \psi}{\partial \rho}\frac{\partial^2 \psi}{\partial \rho^2} &= 4C\sinh\frac{\psi}{2}\cosh\frac{\psi}{2}\frac{\partial \psi}{\partial \rho}, \\
\frac{\partial}{\partial \rho}\left(\frac{\partial \psi}{\partial \rho}\right)^2 &= \frac{\partial}{\partial \rho}\left(2\sqrt{C}\sinh\frac{\psi}{2}\right)^2.
\end{aligned} \tag{A.11}$$

Integrating with respect to boundary condtions yields:

$$\begin{aligned}
\frac{\partial \psi}{\partial \rho} &= -2\sqrt{C}\sinh\frac{\psi}{2} = -4\sinh\frac{\psi}{4}\cosh\frac{\psi}{4}\sqrt{C} = -4\tanh\frac{\psi}{4}\cosh^2\frac{\psi}{4}\sqrt{C}, \\
\frac{1/\tanh\frac{\psi}{4}}{4\cosh^2\frac{\psi}{4}}\frac{\partial \psi}{\partial \rho} &= -\sqrt{C}, \\
\frac{\partial}{\partial \rho}\left(\log\tanh\frac{\psi}{4}\right) &= \frac{\partial}{\partial \rho}\left(-\rho\sqrt{C}\right).
\end{aligned} \tag{A.12}$$

Integrating with respect to boundary condtions yields:

$$\begin{aligned}\log\tanh\frac{\psi}{4} - \log\tanh\frac{\zeta}{4} &= -\rho\sqrt{C},\\ \psi = \varphi - \mathit{\Phi} &= 4\tanh^{-1}\left(\tanh\frac{\zeta}{4}\cdot e^{-\rho\sqrt{C}}\right).\end{aligned} \tag{A.13}$$

A.4 Radial Momentum

Leading-order $O(\delta^{-3})$ radial momentum balance in the Debye layer yields:

$$-\frac{\partial p}{\partial\rho} + \frac{\partial\varphi}{\partial\rho}\frac{\partial^2\varphi}{\partial\rho^2} = 0. \tag{A.14}$$

One integration yields:

$$p = \frac{1}{2}\left(\frac{\partial\varphi}{\partial\rho}\right)^2 = \frac{1}{2}\left(\frac{\partial\psi}{\partial\rho}\right)^2. \tag{A.15}$$

A.5 Tangential Momentum

Leading-order $O(\delta^{-2})$ tangential momentum balance in the Debye layer yields:

$$\begin{aligned}0 &= \frac{\partial^2 v_\theta}{\partial\rho^2} - \frac{\partial p}{\partial\theta} + \frac{\partial\varphi}{\partial\theta}\frac{\partial^2\varphi}{\partial\rho^2},\\ \frac{\partial^2 v_\theta}{\partial\rho^2} &= \frac{1}{2}\frac{\partial}{\partial\theta}\left(\frac{\partial\psi}{\partial\rho}\right)^2 - \frac{\partial(\psi+\mathit{\Phi})}{\partial\theta}\frac{\partial^2\psi}{\partial\rho^2} = \frac{\partial}{\partial\theta}\left(2C\sinh^2\frac{\psi}{2}\right) - \frac{\partial\psi}{\partial\theta}C\sinh\psi - \frac{\partial\mathit{\Phi}}{\partial\theta}\frac{\partial^2\psi}{\partial\rho^2}\\ &= \frac{\partial}{\partial\theta}\left(C\left(\cosh\psi - 1\right)\right) - C\sinh\psi\frac{\partial\psi}{\partial\theta} - \frac{\partial\mathit{\Phi}}{\partial\theta}\frac{\partial^2\psi}{\partial\rho^2}\\ &= \frac{\partial C}{\partial\theta}\left(\cosh\psi - 1\right) - \frac{\partial\mathit{\Phi}}{\partial\theta}\frac{\partial^2\psi}{\partial\rho^2} = \frac{\partial C}{\partial\theta}\left(2\sinh^2\frac{\psi}{2}\right) - \frac{\partial\mathit{\Phi}}{\partial\theta}\frac{\partial^2\psi}{\partial\rho^2}\\ &= -\frac{\partial C}{\partial\theta}\frac{2}{\sqrt{C}}\left(\frac{1}{2}\sinh\frac{\psi}{2}\frac{\partial\psi}{\partial\rho}\right) - \frac{\partial\mathit{\Phi}}{\partial\theta}\frac{\partial^2\psi}{\partial\rho^2} = -\frac{\partial C}{\partial\theta}\frac{2}{\sqrt{C}}\frac{\partial}{\partial\rho}\left(\cosh\frac{\psi}{2}\right) - \frac{\partial\mathit{\Phi}}{\partial\theta}\frac{\partial^2\psi}{\partial\rho^2}.\end{aligned} \tag{A.16}$$

By integrating from specific ρ to $\rho\to\infty$ (where $\psi = 0$):

$$\frac{\partial v_\theta}{\partial\rho} = -\frac{\partial C}{\partial\theta}\frac{2}{\sqrt{C}}\left(\cosh\frac{\psi}{2} - 1\right) - \frac{\partial\mathit{\Phi}}{\partial\theta}\frac{\partial\psi}{\partial\rho} = -\frac{\partial C}{\partial\theta}\frac{4}{\sqrt{C}}\sinh^2\frac{\psi}{4} - \frac{\partial\mathit{\Phi}}{\partial\theta}\frac{\partial\psi}{\partial\rho}. \tag{A.17}$$

Note that:

$$\frac{\partial\psi}{\partial\rho} = -2\sqrt{C}\sinh\frac{\psi}{2} = -4\sqrt{C}\sinh\frac{\psi}{4}\cosh\frac{\psi}{4}, \tag{A.18}$$

$$-4\sinh\frac{\psi}{4} = \frac{1/\cosh\frac{\psi}{4}}{\sqrt{C}}\frac{\partial\psi}{\partial\rho}. \tag{A.19}$$

Thus:

$$\frac{\partial v_\theta}{\partial \rho} = \left(\frac{\partial C}{\partial \theta}\frac{1}{C}\right)\cdot\left(\frac{\sinh\frac{\psi}{4}}{\cosh\frac{\psi}{4}}\frac{\partial \psi}{\partial \rho}\right) - \frac{\partial \varPhi}{\partial \theta}\frac{\partial \psi}{\partial \rho} = 4\frac{\partial}{\partial \rho}\left(\log\cosh\frac{\psi}{4}\right)\frac{\partial}{\partial \theta}\log C - \frac{\partial \psi}{\partial \rho}\frac{\partial \varPhi}{\partial \theta}, \tag{A.20}$$

$$\frac{\partial v_\theta}{\partial \rho} = \frac{\partial}{\partial \rho}\left(4\log\cosh\frac{\psi}{4}\frac{\partial}{\partial \theta}\log C - \psi\frac{\partial \varPhi}{\partial \theta}\right). \tag{A.21}$$

By integrating from $\rho = 0$ to $\rho \to \infty$, we have Dukhin-Derjaguin slip formula for tangential velocity component V_θ (where the radial component is $V_r = 0$) for $\zeta = \mathscr{V} - \varPhi$:

$$V_\theta = \zeta\cdot\frac{\partial \varPhi}{\partial \theta} - 4\log\cosh\frac{\zeta}{4}\cdot\frac{\partial}{\partial \theta}\log C = \zeta\cdot\frac{\partial \varPhi}{\partial \theta} + 2\log\left(1-\tanh^2\frac{\zeta}{4}\right)\cdot\frac{\partial}{\partial \theta}\log C. \tag{A.22}$$

Appendix B

Differential operators in spherical coordinates

The following spherical coordinate system is used:

$$\begin{aligned} x &= r\sin\theta\cos\phi \\ y &= r\sin\theta\sin\phi \\ z &= r\cos\theta \end{aligned} \tag{B.1}$$

Assume full symmetry around z axis – there is no dependence on ϕ:

$$\boldsymbol{\nabla} f = \frac{\partial f}{\partial r}\hat{\mathbf{r}} + \frac{1}{r}\frac{\partial f}{\partial \theta}\hat{\theta} + \frac{1}{r\sin\theta}\frac{\partial f}{\partial \phi}\hat{\phi} \tag{B.2}$$

$$\boldsymbol{\nabla}\cdot\mathbf{F} = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\cdot F_r\right) + \frac{1}{r\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\cdot F_\theta\right) + \frac{1}{r\sin\theta}\frac{\partial F_\phi}{\partial\phi} \tag{B.3}$$

B.1 Scalar Operators

Scalar Laplacian derivation:

$$\nabla^2 f = \boldsymbol{\nabla}\cdot\boldsymbol{\nabla} f = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial f}{\partial r}\right) + \frac{1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\cdot\frac{\partial f}{\partial\theta}\right) + \frac{1}{r^2\sin^2\theta}\frac{\partial^2 f}{\partial\phi^2} \tag{B.4}$$

In conservative form:

$$r^2\sin\theta\cdot\boldsymbol{\nabla}^2 f = \frac{\partial}{\partial r}\left(r^2\sin\theta\frac{\partial f}{\partial r}\right) + \frac{\partial}{\partial\theta}\left(\sin\theta\cdot\frac{\partial f}{\partial\theta}\right) + \frac{\partial}{\partial\phi}\left(\frac{1}{\sin\theta}\cdot\frac{\partial f}{\partial\phi}\right) \tag{B.5}$$

The unit vectors in spherical coordinate system are:

$$\hat{\boldsymbol{r}} = \frac{\partial \boldsymbol{r}}{\partial r} = \begin{pmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{pmatrix}, \quad \hat{\boldsymbol{\theta}} = \frac{1}{r}\frac{\partial \boldsymbol{r}}{\partial \theta} = \begin{pmatrix} \cos\theta\cos\phi \\ \cos\theta\sin\phi \\ -\sin\theta \end{pmatrix}, \quad \hat{\boldsymbol{\phi}} = \frac{1}{r\sin\theta}\frac{\partial \boldsymbol{r}}{\partial \phi} = \begin{pmatrix} -\sin\phi \\ \cos\phi \\ 0 \end{pmatrix} \tag{B.6}$$

Note that:

$$\begin{aligned} \frac{\partial \hat{\boldsymbol{r}}}{\partial r} = \boldsymbol{0}, \frac{\partial \hat{\boldsymbol{\theta}}}{\partial r} = \boldsymbol{0}, \frac{\partial \hat{\boldsymbol{\phi}}}{\partial r} = \boldsymbol{0}, \\ \frac{\partial \hat{\boldsymbol{r}}}{\partial \theta} = \hat{\boldsymbol{\theta}}, \frac{\partial \hat{\boldsymbol{\theta}}}{\partial \theta} = -\hat{\boldsymbol{r}}, \frac{\partial \hat{\boldsymbol{\phi}}}{\partial \theta} = \boldsymbol{0}, \\ \frac{\partial \hat{\boldsymbol{r}}}{\partial \phi} = \sin\theta \cdot \hat{\boldsymbol{\phi}}, \frac{\partial \hat{\boldsymbol{\theta}}}{\partial \phi} = \cos\theta \cdot \hat{\boldsymbol{\phi}}, \frac{\partial \hat{\boldsymbol{\phi}}}{\partial \phi} = -\sin\theta \cdot \hat{\boldsymbol{r}} - \cos\theta \cdot \hat{\boldsymbol{\theta}}. \end{aligned} \tag{B.7}$$

B.2 Vector Operators

Vector gradient operator can be written by:

$$\begin{aligned} &\boldsymbol{\nabla V} = \left\{ \hat{\boldsymbol{r}}\frac{\partial}{\partial r} + \hat{\boldsymbol{\theta}}\frac{1}{r}\frac{\partial}{\partial \theta} + \hat{\boldsymbol{\phi}}\frac{1}{r\sin\theta}\frac{\partial}{\partial \phi} \right\} \boldsymbol{V} = \left\{ \hat{\boldsymbol{r}}\frac{\partial}{\partial r} + \hat{\boldsymbol{\theta}}\frac{1}{r}\frac{\partial}{\partial \theta} + \hat{\boldsymbol{\phi}}\frac{1}{r\sin\theta}\frac{\partial}{\partial \phi} \right\} \left(V_r\hat{\boldsymbol{r}} + V_\theta\hat{\boldsymbol{\theta}} + V_\phi\hat{\boldsymbol{\phi}} \right) \\ &\frac{\partial \boldsymbol{V}}{\partial r} = \frac{\partial V_r}{\partial r}\hat{\boldsymbol{r}} + \frac{\partial V_\theta}{\partial r}\hat{\boldsymbol{\theta}} + \frac{\partial V_\phi}{\partial r}\hat{\boldsymbol{\phi}} \\ &\frac{\partial \boldsymbol{V}}{\partial \theta} = \frac{\partial V_r}{\partial \theta}\hat{\boldsymbol{r}} + \frac{\partial V_\theta}{\partial \theta}\hat{\boldsymbol{\theta}} + \frac{\partial V_\phi}{\partial \theta}\hat{\boldsymbol{\phi}} + V_r\hat{\boldsymbol{\theta}} - V_\theta\hat{\boldsymbol{r}} \\ &\frac{\partial \boldsymbol{V}}{\partial \phi} = \frac{\partial V_r}{\partial \phi}\hat{\boldsymbol{r}} + \frac{\partial V_\theta}{\partial \phi}\hat{\boldsymbol{\theta}} + \frac{\partial V_\phi}{\partial \phi}\hat{\boldsymbol{\phi}} + V_r\sin\theta\hat{\boldsymbol{\phi}} + V_\theta\cos\theta\hat{\boldsymbol{\phi}} - V_\phi\left(\hat{\boldsymbol{r}}\sin\theta + \hat{\boldsymbol{\theta}}\cos\theta\right) \\ &\boldsymbol{\nabla V} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{r} & 0 \\ 0 & 0 & \frac{1}{r\sin\theta} \end{pmatrix} \begin{pmatrix} \frac{\partial V_r}{\partial r} & \frac{\partial V_\theta}{\partial r} & \frac{\partial V_\phi}{\partial r} \\ \frac{\partial V_r}{\partial \theta} - V_\theta & \frac{\partial V_\theta}{\partial \theta} + V_r & \frac{\partial V_\phi}{\partial \theta} \\ \frac{\partial V_r}{\partial \phi} - V_\phi\sin\theta & \frac{\partial V_\theta}{\partial \phi} - V_\phi\cos\theta & \frac{\partial V_\phi}{\partial \phi} + V_r\sin\theta + V_\theta\cos\theta \end{pmatrix} \end{aligned} \tag{B.8}$$

Vector Laplacian components can be written as:

$$\nabla^2\boldsymbol{V} = \boldsymbol{\nabla}\cdot\boldsymbol{\nabla V} = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial \boldsymbol{V}}{\partial r}\right) + \frac{1}{r^2\sin\theta}\frac{\partial}{\partial \theta}\left(\sin\theta\frac{\partial \boldsymbol{V}}{\partial \theta}\right) + \frac{1}{r^2\sin^2\theta}\frac{\partial}{\partial \phi}\left(\frac{\partial \boldsymbol{V}}{\partial \phi}\right) \tag{B.9}$$

$\hat{\boldsymbol{r}}$ component:

$$\begin{aligned}
\nabla^2(V_r\hat{\boldsymbol{r}}) &= \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial V_r}{\partial r}\right)\hat{\boldsymbol{r}} \\
&+ \frac{1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\left(\frac{\partial V_r}{\partial\theta}\hat{\boldsymbol{r}} + V_r\hat{\boldsymbol{\theta}}\right)\right) \\
&+ \frac{1}{r^2\sin^2\theta}\frac{\partial}{\partial\phi}\left(\frac{\partial V_r}{\partial\phi}\hat{\boldsymbol{r}} + V_r\sin\theta\hat{\boldsymbol{\phi}}\right) \\
\nabla^2(V_r\hat{\boldsymbol{r}}) &= \left(\frac{2}{r}\frac{\partial V_r}{\partial r} + \frac{\partial^2 V_r}{\partial r^2}\right)\hat{\boldsymbol{r}} \qquad \text{(B.10)} \\
&+ \frac{1}{r^2\sin\theta}\left(\cos\theta\left(\frac{\partial V_r}{\partial\theta}\hat{\boldsymbol{r}} + V_r\hat{\boldsymbol{\theta}}\right) + \sin\theta\left(\frac{\partial^2 V_r}{\partial\theta^2}\hat{\boldsymbol{r}} + \frac{\partial V_r}{\partial\theta}\hat{\boldsymbol{\theta}} + \frac{\partial V_r}{\partial\theta}\hat{\boldsymbol{\theta}} - V_r\hat{\boldsymbol{r}}\right)\right) \\
&+ \frac{1}{r^2\sin^2\theta}\left(\frac{\partial^2 V_r}{\partial\phi^2}\hat{\boldsymbol{r}} + \frac{\partial V_r}{\partial\phi}\sin\theta\hat{\boldsymbol{\phi}} + \frac{\partial V_r}{\partial\phi}\hat{\boldsymbol{\phi}}\sin\theta - V_r\sin\theta\left(\hat{\boldsymbol{r}}\sin\theta + \hat{\boldsymbol{\theta}}\cos\theta\right)\right) \\
\nabla^2(V_r\hat{\boldsymbol{r}}) &= \left(\frac{2}{r}\frac{\partial V_r}{\partial r} + \frac{\partial^2 V_r}{\partial r^2} - \frac{2V_r}{r^2} + \frac{1}{r^2}\frac{\partial^2 V_r}{\partial\theta^2} + \frac{\cot\theta}{r^2}\frac{\partial V_r}{\partial\theta} + \frac{1}{r^2\sin^2\theta}\frac{\partial^2 V_r}{\partial\phi^2}\right)\hat{\boldsymbol{r}} \\
&+ \frac{2}{r^2}\frac{\partial V_r}{\partial\theta}\hat{\boldsymbol{\theta}} + \frac{2}{r^2\sin\theta}\frac{\partial V_r}{\partial\phi}\hat{\boldsymbol{\phi}}
\end{aligned}$$

$\hat{\boldsymbol{\theta}}$ component:

$$\begin{aligned}
\nabla^2(V_\theta\hat{\boldsymbol{\theta}}) &= \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial V_\theta}{\partial r}\right)\hat{\boldsymbol{\theta}} \\
&+ \frac{1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\left(\frac{\partial V_\theta}{\partial\theta}\hat{\boldsymbol{\theta}} - V_\theta\hat{\boldsymbol{r}}\right)\right) \\
&+ \frac{1}{r^2\sin^2\theta}\frac{\partial}{\partial\phi}\left(\frac{\partial V_\theta}{\partial\phi}\hat{\boldsymbol{\theta}} + V_\theta\cos\theta\hat{\boldsymbol{\phi}}\right) \\
\nabla^2(V_\theta\hat{\boldsymbol{\theta}}) &= \left(\frac{2}{r}\frac{\partial V_\theta}{\partial r} + \frac{\partial^2 V_\theta}{\partial r^2}\right)\hat{\boldsymbol{\theta}} \qquad \text{(B.11)} \\
&+ \frac{1}{r^2\sin\theta}\left(\cos\theta\left(\frac{\partial V_\theta}{\partial\theta}\hat{\boldsymbol{\theta}} - V_\theta\hat{\boldsymbol{r}}\right) + \sin\theta\left(\frac{\partial^2 V_\theta}{\partial\theta^2}\hat{\boldsymbol{\theta}} - \frac{\partial V_\theta}{\partial\theta}\hat{\boldsymbol{r}} - \frac{\partial V_\theta}{\partial\theta}\hat{\boldsymbol{r}} - V_\theta\hat{\boldsymbol{\theta}}\right)\right) \\
&+ \frac{1}{r^2\sin^2\theta}\left(\frac{\partial^2 V_\theta}{\partial\phi^2}\hat{\boldsymbol{\theta}} + \frac{\partial V_\theta}{\partial\phi}\cos\theta\hat{\boldsymbol{\phi}} + \frac{\partial V_\theta}{\partial\phi}\cos\theta\hat{\boldsymbol{\phi}} - V_\theta\cos\theta\left(\hat{\boldsymbol{r}}\sin\theta + \hat{\boldsymbol{\theta}}\cos\theta\right)\right) \\
\nabla^2(V_\theta\hat{\boldsymbol{\theta}}) &= \left(-\frac{2\cot\theta}{r^2}V_\theta - 2\frac{\partial V_\theta}{\partial\theta}\right)\hat{\boldsymbol{r}} + \left(\frac{2}{r}\frac{\partial V_\theta}{\partial r} + \frac{\partial^2 V_\theta}{\partial r^2} + \frac{\cot\theta}{r^2}\frac{\partial V_\theta}{\partial\theta} + \frac{1}{r^2}\frac{\partial^2 V_\theta}{\partial\theta^2}\right)\hat{\boldsymbol{\theta}} \\
&+ \frac{1}{r^2\sin^2\theta}\left(\frac{\partial^2 V_\theta}{\partial\phi^2} - V_\theta\right)\hat{\boldsymbol{\theta}} + \frac{2\cot\theta}{r^2\sin\theta}\frac{\partial V_\theta}{\partial\phi}\hat{\boldsymbol{\phi}}
\end{aligned}$$

$\hat{\boldsymbol{\phi}}$ component:

$$\begin{aligned}
\nabla^2(V_\phi\hat{\boldsymbol{\phi}}) &= \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial V_\phi}{\partial r}\right)\hat{\boldsymbol{\phi}} + \frac{1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial V_\phi}{\partial\theta}\hat{\boldsymbol{\phi}}\right)\\
&\quad + \frac{1}{r^2\sin^2\theta}\frac{\partial}{\partial\phi}\left(\frac{\partial V_\phi}{\partial\phi}\hat{\boldsymbol{\phi}} - V_\phi\left(\hat{\boldsymbol{r}}\sin\theta + \hat{\boldsymbol{\theta}}\cos\theta\right)\right)\\
\nabla^2(V_\phi\hat{\boldsymbol{\phi}}) &= \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial V_\phi}{\partial r}\right)\hat{\boldsymbol{\phi}} + \frac{1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial V_\phi}{\partial\theta}\hat{\boldsymbol{\phi}}\right) \qquad \text{(B.12)}\\
&\quad + \frac{1}{r^2\sin^2\theta}\frac{\partial}{\partial\phi}\left(\frac{\partial V_\phi}{\partial\phi}\hat{\boldsymbol{\phi}} - V_\phi\left(\hat{\boldsymbol{r}}\sin\theta + \hat{\boldsymbol{\theta}}\cos\theta\right)\right)\\
\nabla^2(V_\phi\hat{\boldsymbol{\phi}}) &= \left(\frac{2}{r}\frac{\partial V_\phi}{\partial r} + \frac{\partial^2 V_\phi}{\partial r^2}\right)\hat{\boldsymbol{\phi}} + \frac{1}{r^2\sin\theta}\left(\cos\theta\frac{\partial V_\phi}{\partial\theta} + \sin\theta\frac{\partial^2 V_\phi}{\partial\theta^2}\right)\hat{\boldsymbol{\phi}}\\
&\quad + \frac{1}{r^2\sin^2\theta}\left(\frac{\partial^2 V_\phi}{\partial\phi^2}\hat{\boldsymbol{\phi}} - 2\frac{\partial V_\phi}{\partial\phi}\left(\hat{\boldsymbol{r}}\sin\theta + \hat{\boldsymbol{\theta}}\cos\theta\right) - V_\phi\hat{\boldsymbol{\phi}}\right)
\end{aligned}$$

In summary:

$$\begin{aligned}
\nabla^2\boldsymbol{V} &= \left(\nabla^2 V_r - \frac{2V_r}{r^2}\right)\hat{\boldsymbol{r}} + \frac{2}{r^2}\frac{\partial V_r}{\partial\theta}\hat{\boldsymbol{\theta}} + \frac{2}{r^2\sin\theta}\frac{\partial V_r}{\partial\phi}\hat{\boldsymbol{\phi}}\\
&\quad - \frac{2}{r^2\sin\theta}\frac{\partial\left(V_\theta\sin\theta\right)}{\partial\theta}\hat{\boldsymbol{r}} + \left(\nabla^2 V_\theta - \frac{V_\theta}{r^2\sin^2\theta}\right)\hat{\boldsymbol{\theta}} + \frac{2\cot\theta}{r^2\sin\theta}\frac{\partial V_\theta}{\partial\phi}\hat{\boldsymbol{\phi}} \qquad \text{(B.13)}\\
&\quad - \frac{2}{r^2\sin\theta}\frac{\partial V_\phi}{\partial\phi}\hat{\boldsymbol{r}} - \frac{2\cot\theta}{r^2\sin\theta}\frac{\partial V_\phi}{\partial\phi}\hat{\boldsymbol{\theta}} + \left(\nabla^2 V_\phi - \frac{V_\phi}{r^2\sin^2\theta}\right)\hat{\boldsymbol{\phi}}
\end{aligned}$$

Since $V_\phi = 0$ and $\frac{\partial}{\partial\phi}(\cdot) = 0$, we get:

$$\begin{aligned}
\nabla^2\boldsymbol{V} &= \left(\nabla^2 V_r - \frac{2V_r}{r^2} - \frac{2}{r^2\sin\theta}\frac{\partial\left(V_\theta\sin\theta\right)}{\partial\theta}\right)\hat{\boldsymbol{r}} + \left(\nabla^2 V_\theta - \frac{V_\theta}{r^2\sin^2\theta} + \frac{2}{r^2}\frac{\partial V_r}{\partial\theta}\right)\hat{\boldsymbol{\theta}}\\
\nabla^2\boldsymbol{V} &= \left(\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial V_r}{\partial r}\right) + \frac{1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\cdot\frac{\partial V_r}{\partial\theta}\right) - \frac{2V_r}{r^2} - \frac{2}{r^2\sin\theta}\frac{\partial\left(V_\theta\sin\theta\right)}{\partial\theta}\right)\hat{\boldsymbol{r}} \qquad \text{(B.14)}\\
&\quad + \left(\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial V_\theta}{\partial r}\right) + \frac{1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\cdot\frac{\partial V_\theta}{\partial\theta}\right) - \frac{V_\theta}{r^2\sin^2\theta} + \frac{2}{r^2}\frac{\partial V_r}{\partial\theta}\right)\hat{\boldsymbol{\theta}}
\end{aligned}$$

Bibliography

[1] E. Yariv. Migration of ion-exchange particles driven by a uniform electric field. *Journal of Fluid Mechanics*, 655:1–17, 2010.

[2] O. Schnitzer and E. Yariv. Macroscale description of electrokinetic flows at large zeta potentials: Nonlinear surface conduction. *Phys. Rev. E*, 86:021503, Aug 2012.

[3] J. H. Masliyah and S. Bhattacharjee. *Electrokinetic and Colloid Transport Phenomena*. John Wiley & Sons, Inc., 2005.

[4] B. Kirby. *Micro- and Nanoscale Fluid Mechanics – Transport in Microfluidic Devices*. Cambridge Univ Press, 2010.

[5] E. Yariv. An asymptotic derivation of the thin-Debye-layer limit for electrokinetic phenomena. *Chemical Engineering Communications*, 197(1):3–17, 2010.

[6] G.M. Whitesides. The origins and the future of microfluidics. *Nature*, 442(7101):368–373, 2006.

[7] D. Erickson and D. Li. Integrated microfluidic devices. *Analytica Chimica Acta*, 507(1):11–26, 2004.

[8] T.M. Squires and S.R. Quake. Microfluidics: Fluid physics at the nanoliter scale. *Reviews of modern physics*, 77(3):977, 2005.

[9] S. Hardt and F. Schönfeld. *Microfluidic technologies for miniaturized analysis systems*, volume 18. Springer, 2007.

[10] A. Wainright, U.T. Nguyen, T.L. Bjornson, and T.D. Boone. Preconcentration and separation of double-stranded dna fragments by electrophoresis in plastic microfluidic devices. *Electrophoresis*, 24(21):3784–3792, 2003.

[11] H. Wu, A. Wheeler, and R.N. Zare. Chemical cytometry on a picoliter-scale integrated microfluidic chip. *Proceedings of the National Academy of Sciences of the United States of America*, 101(35):12809–12813, 2004.

[12] R. Weinberger. *Practical capillary electrophoresis*. Academic Press San Diego, CA:, 1993.

[13] K.M. Horsman, J.M. Bienvenue, K.R. Blasier, and J.P. Landers. Forensic dna analysis on microfluidic devices: a review. *Journal of forensic sciences*, 52(4):784–799, 2007.

[14] MV Kothare. Dynamics and control of integrated microchemical systems with application to micro-scale fuel processing. *Computers & chemical engineering*, 30(10):1725–1734, 2006.

[15] F. C. Leinweber and U. Tallarek. Nonequilibrium electrokinetic effects in beds of ion-permselective particles. *Langmuir*, 20(26):11637–11648, 2004. PMID: 15595793.

[16] M.W. Losey, M.A. Schmidt, and K.F. Jensen. Microfabricated multiphase packed-bed reactors: Characterization of mass transfer and reactions. *Industrial & engineering chemistry research*, 40(12):2555–2562, 2001.

[17] S.J. Kim, S.H. Ko, K.H. Kang, and J. Han. Direct seawater desalination by ion concentration polarization. *Nature Nanotechnology*, 5(4):297–301, 2010.

[18] J.T. Santini Jr, A.C. Richards, R. Scheidt, M.J. Cima, and R. Langer. Microchips as controlled drug-delivery devices. *Angewandte Chemie International Edition*, 39(14):2396–2407, 2000.

[19] A. Bard and L Faulkner. *Electrochemical Methods - Fundamentals and Applications*. Wiley, 2001.

[20] Shau-Chun Wang, Yi-Wen Lai, Yuxing Ben, and Hsueh-Chia Chang. Microfluidic mixing by dc and ac nonlinear electrokinetic vortex flows. *Industrial & Engineering Chemistry Research*, 43(12):2902–2911, 2004.

[21] Y. Ben and H.-C. Chang. Nonlinear smoluchowski slip velocity and micro-vortex generation. *Journal of Fluid Mechanics*, 461:229–238, 2002.

[22] J.H. Ferziger and J.H.F.M. Perić. *Computational Methods for Fluid Dynamics*. Springer-Verlag GmbH, 2002.

[23] J. Strikwerda. *Finite difference schemes and partial differential equations*. Society for Industrial Mathematics, 2004.

[24] A. Sidi. *Practical extrapolation methods: theory and applications*, volume 10. Cambridge University Press, 2003.

[25] R. Zeyde. Electrokinetic Solver Source Code. http://github.com/romanz/thesis.

[26] Timothy A. Davis. Algorithm 832: Umfpack v4.3—an unsymmetric-pattern multifrontal method. *ACM Trans. Math. Softw.*, 30(2):196–199, June 2004.

[27] O. Schnitzer, R. Zeyde, I. Yavneh, and E. Yariv. Non-linear electrophoresis of a highly charged colloidal particle. *Submitted to Phyiscs of fluids*, 2013.

[28] Technion Taub Computer Center. TAMNUN HPC cluster. http://hpc.technion.ac.il/.

[29] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.

אלקטרוקינטיקה חישובית

רומן זיידה

אלקטרוקינטיקה חישובית

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
מגיסטר למדעים במדעי המחשב

רומן זיידה

הוגש לסנט הטכניון - מכון טכנולוגי לישראל
אדר ה'תשע"ג   חיפה   מרץ 2013

המחקר נעשה בהנחיית פרופ' עירד יבנה בפקולטה למדעי המחשב.

תודות

ברצוני להודות למנחה שלי, פרופ' עירד יבנה, על ההנחייה היוצאת מן הכלל, הדיונים המועילים והעזרה הרבה במהלך הדרך, וכן למשפחתי, ובמיוחד לאשתי, גלית, על אהבתכם ולתמיכתכם.

אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי.

תקציר

מבוא

בעבודה זו אנו חוקרים את התנועה האלקטרוקינטית של חלקיקים בתמיסת אלקטרוליט תחת השפעה של שדה חשמלי, ומפתחים תשתית נומרית לפתרון איטרטיבי של בעיות פיזיקליות לא-לינאריות מסוג זה. באמצעות תהליכים אלקטרוקינטיים ניתן להניע ולבצע מניפולציות בחלקיקים זעירים בתחומים שונים, כגון: ננו-טכנולוגיה, זרימה זעירה, הפרדה וניקוי, התפלת מים, אלקטרופורזה ועוד. בשל כוחות אלקטרוסטטיים חזקים, נוצרת שכבת שפה דקה קרוב לפני החלקיק. עקב הפער בין סקאלת החלקיק לבין סקאלת שכבת השפה, נוצר קושי להגיע לפתרון נומרי של הבעיה המלאה.

אנו עושים שימוש בתנאי שפה אפקטיביים המתקבלים מאנליזה אסימפטוטית של שכבת השפה והתכונות הכימיות של החלקיק. תנאי שפה אלה פותחו על ידי פרופ' אהוד יריב ואורי שניצר עבור מחליפי יונים וחלקיקים אינרטיים בעלי מטען שפה, ובהם אנו משתמשים לצורך בניית פותר נומרי לפתרון הבעיה הלא-לינארית בסקאלת החלקיק. אנו מבצעים דיסקרטיזציה של המשוואות הדיפרנציאליות ושל תנאי השפה האפקטיביים, ומפתחים תשתית נומרית אשר מאפשרת בנייה אוטומטית של פותר איטרטיבי בשיטת ניוטון. הפותר מאפשר קבלת תוצאות נומריות לתנועת חלקיקים מהסוג שלעיל, בהינתן עוצמת השדה החשמלי המופעל. התוצאות עבור שדות חלשים אומתו על ידי השוואה לפתרונות אסימפטוטיים של הבעיה. כמו כן, קיבלנו באמצעות הפותר הנומרי לראשונה פתרונות בשדות חזקים, אשר אין להם עדיין קירובים אסימפטוטיים. מפתרונות אלה ניתן ללמוד על אופי המערכת בשדות חזקים, כולל התנהגות אסימפטוטית של שכבת שפה נוספת אשר נוצרת בחזית החלקיק.

הבעיה הפיזיקלית

התיאוריה האלקטרוקינטית מתארת את הדינמיקה של חלקיקים טעונים חשמלית בתוך תמיסת יונים. כאשר חלקיק צובר מטען על פני שפתו, נוצרת שכבת יונים בעלת מטען נגדי בעקבות כוחות משיכה חשמליים וכך נוצרת מבנה של שכבה כפולה סביב לחלקיק. שכבה זו, הנקראת ״שכבת דבאי״, ממסכת את מטען השפה של החלקיק ויוצרת עקב כך הפרש פוטנציאל בין שפת החלקיק לבין הנוזל הניטרלי מחוץ לשכבה. במקרה בו רוחב השכבה קטן באופן משמעותי מאשר ממדי החלקיק, ניתן לפתח פתרון אנליטי אסימפטוטי של הדינמיקה בשכבת השפה.

המשתנים בבעיית זרימה אלקטרוקינטית הם הפוטנציאל החשמלי, מהירות הנוזל והלחץ, וריכוז היונים. תנאי השפה מוגדרים בהתאם לתנאי הבעיה הספציפית ונקבעים על ידי צורת החלקיק, תכונותיו הכימיות והדינמיקה של הנוזל. המשוואות הדיפרנציאליות החלקיות אשר מתארות את התנהגות המערכת תחת השפעתו של שדה חשמלי חיצוני הן מצומדות, לא-לינאריות, וללא פתרון אנליטי ידוע. כמו כן, על כל פותר נומרי יהיה להתמודד עם פער הסקאלות שנוצר כתוצאה מכך שעובי שכבת השפה קטן משמעותית לעומת ממדי החלקיק.

על מנת להתמודד עם קושי זה, ניתן לפתח תנאי שפה אפקטיביים מחוץ לשכבת דבאי ולקבל מודל מקרוסקופי לבעיה על ידי שימוש בפתרון האסימפטוטי לשכבת השפה שלעיל. מודל זה ניתן לפתרון מקורב עבור חלקיק כדורי שנמצא תחת השפעה של שדה חשמלי חלש, אך קשה להרחיב פתרון אנליטי זה עבור מקרים כלליים יותר. יש לציין כי כאשר השדה החשמלי מתחזק, צפויות להתרחש תופעות לא-לינאריות משמעותיות, אשר עדיין לא נחקרו באופן נרחב.

שימושים

לתופעות האלקטרוקינטיות שתוארו לעיל נמצאו שימושים רבים בפיתוח התקנים למיקרו-זרימה ושיטות לבקרה ומניפולציה של זרימת נוזלים בסקאלות של מיקרונים וננומטרים. לצורך כך, נדרש מחקר ופיתוח של רכיבי מיקרו-זרימה שונים, כגון שסתומים, משאבות, גלאים, מיקסרים, מסננים, התקנים להפרדת חומרים וכדומה. רכיבים אלו מורכבים על מערכות זעירות הנקראות "מעבדה-על-שבב", ומחוברים על ידי מיקרו-תעלות. מערכות אלו מיוצרות בשיטות של ננו-ליטוגרפיה. ניתן להשפיע על אופי הזרימה על ידי הפעלת שדה חשמלי חיצוני על פני המערכת כולה, או לחלופין, על ידי הפעלתו באופן מקומי בתוך מיקרו-תעלה ספציפית. לרוב, מספר ריינולדס בזרימה כזו הוא נמוך מאוד. על כן, האפקט הדומיננטי בזרימה זו הוא הצמיגות, ואיבר ההסעה של משוואות נבייה-סטוקס הופך להיות זניח, ולפיכך, מיקרו-זרימה היא לרוב זרימה למינארית.

בעשור האחרון התרחשה התקדמות משמעותית בחקר מיקרו-זרימה, הודות לתפוצתם של כלים ושיטות לייצור של מערכות קטנות, זולות וניידות עבור אבחון רפואי מהיר, ועבור מחקר בסיסי של תהליכים פיזיקליים, כימיים וביולוגיים. ניתן למנות שימושים לדוגמה וביניהם הפרדה של די.אנ.איי, אנליזה של תאים וזיהוי פלילי, מערכות מיקרו-כימיות מורכבות ומשולבות, הפרדה מבוססת מצע צפוף, התפלת מים, העברה פיזיולוגית של תרופות אל רקמת המטרה, זיהוי גורמי מחלה ועוד.

מטרת העבודה

מטרת עבודה זו הינה לפתח ולממש תשתית תוכנה לצורך ייצור של פותרים נומריים איטרטיביים עבור בעיות אלקטרוקינטיות מקרוסקופיות לא-לינאריות, וליישם תשתית זו לתיאור ומחקר של מערכות אשר אין להן פתרונות אנליטיים ידועים. שתי מערכות בהן מתמקדת עבודה זו הן נדידת חלקיק מחליף יונים בשדה חשמלי ואלקטרופורזה של חלקיק אינרטי בעל מטען שפה. התשתית מורכבת מאוסף של מודולים המאפשרים לבנות מודל דיסקרטי של בעיה אלקטרוקינטית ספציפית. בהינתן מודל זה, התשתית מאפשרת לפתור את מערכת המשוואות הלא-לינארית באופן אוטומטי. כך ניתן לחקור מערכות שונות מבחינה פיזיקלית ללא שינוי בקוד של הפותר הנומרי. התשתית הנומרית והפתרונות הנומריים המתקבלים ממנה מאפשרים לקבל תובנות על התהליכים הכימיים והפיזיקליים במקרים הרבה יותר כלליים מן הידוע כיום. כמו כן, התוצאות של הפותר צפויות לקדם פיתוח אנליטי של פתרונות אסימפטוטיים מקורבים, על סמך ההתנהגות האסימפטוטית של הפתרונות הנומריים. כמו כן, הפותר הנומרי יכול לשמש למחקר תיאורטי של תופעות פיזיקליות מעניינות שנצפו באופן ניסיוני. תופעה אחת שניתן לציין כדוגמה היא הופעה של מיקרו-מערבולות בזרימה אלקטרוקינטית לא-לינארית, אשר יכולה לשמש כמיקסר עבור אפליקציות שונות של מיקרו-זרימה.

השיטה הנומרית

על מנת לבנות פותר נומרי לבעיה אלקטרוקינטית ספציפית, יש לבצע תחילה דיסקרטיזציה של מערכת המשוואות הדיפרנציאליות החלקיות ושל תנאי השפה על פני סריג סופי. אנו עושים שימוש בסריג רגולרי במערכת קואורדינטות כדורית, על מנת לנצל את הסימטריה הצירית בבעיה בתיאור תנאי השפה וחיסכון בזיכרון הנדרש לפותר, מכיוון שהבעיה הופכת להיות "דו-מימדית". מערכת המשוואות הדיסקרטית היא לא לינארית ולכן אנו עושים שימוש בפותר איטרטיבי מסוג ניוטון על מנת לקבל את הפתרון הדיסקרטי אחרי מספר קטן של איטרציות. על מנת לעשות שימוש בפותר זה, יש לנסח את מערכת המשוואות כאופרטור רב-ממדי, אשר עבורו אנו נדרשים למצוא את וקטור הקלט שיאפס את תוצאתו. בשיטת ניוטון, אנו נדרש לחשב את הגרדיינט של אופרטור זה על מנת לקרב אותו על ידי אופרטור לינארי וכך לחשב את השינוי הנדרש לווקטור הקלט, באמצעות פתרון של מערכת משוואות לינארית, המתקבלת מהגרדיינט ותוצאת האופרטור הנוכחיים.

מכיוון שחישוב אנליטי של הגרדיינט הוא מסובך למדי עבור הבעיות אותן אנו באים לפתור, בחרנו לממש את חישוב הגרדיינט על ידי שיטה של גזירה אוטומטית. בשיטה זו, האופרטור הלא-לינארי, אשר מייצג את מערכת המשוואות שיש לפתור, נוצר על ידי אלגברה של אופרטורים בסיסיים ופשוטים יותר, כגון אופרטור קבוע, וקטור הקלט, הפעלת טרנספורמציה לינארית, חישוב של פונקציה אנליטית, חיבור, חיסור, כפל איבר-איבר והרכבה של אופרטורים אחרים. כל אופרטור מממש את הממשק הבסיסי, המאפשר את חישוב וקטור התוצאה ואת מטריצת הגרדיינט שלו, בהינתן וקטור הקלט. החישוב מתאפשר על ידי הפעלה של כלל השרשרת בגזירה של אופרטורים מורכבים, כך שכל המידע שאופרטור נדרש על מנת לחשב את הגרדיינט שלו, מתקבל מהאופרטורים שמרכיבים אותו באופן היררכי. מכיוון שהגרדיינטים המתקבלים ניתנים לייצוג יעיל על ידי מטריצות דלילות, המשאבים הנדרשים לחישוב הגרדיינט בשיטה זו זניחים יחסית למשאבים הנדרשים לפתרון המערכת הלינארית הנדרשת לחישוב צעד ניוטון. מבנה נתונים זה מאפשר לבצע אופטימיזציה על הייצוג האופרטורי של מערכת המשוואות. כמו כן, נציין כי שינויים במודל הפיזיקלי של המערכת אינם דורשים שינויים במימוש הנומרי של הפותר, אלא רק בשינוי של האופרטורים הרלוונטיים בייצוג הדיסקרטי של הבעיה.