A conservative discretization of the Poisson–Nernst–Planck equations on adaptive Cartesian grids

Mohammad Mirzadeh a,*, Frédéric Gibou a, b

a Department of Mechanical Engineering, University of California, Santa Barbara, CA 93106, United States
b Department of Computer Science, University of California, Santa Barbara, CA 93106, United States

A R T I C L E   I N F O
Article history:
Received 17 September 2013
Received in revised form 6 June 2014
Accepted 19 June 2014
Available online 27 June 2014

Keywords:
Nonlinear Poisson–Nernst–Planck equation
Non-graded adaptive grid
Quadtree data structure
Second-order discretization
Arbitrary geometries
Supercapacitors
Level-set

A B S T R A C T
In this paper we present a novel hybrid finite-difference/finite-volume method for the numerical solution of the nonlinear Poisson–Nernst–Planck (PNP) equations on irregular domains. The method is described in two spatial dimensions but can be extended to three dimensional problems as well. The boundary of the irregular domain is represented implicitly via the zero level set of a signed distance function and quadtree data structures are used to systematically generate adaptive grids needed to accurately capture the electric double layer near the boundary. To handle the nonlinearity in the PNP equations efficiently, a semi-implicit time integration method is utilized. An important feature of our method is that total number of ions in the system is conserved by carefully imposing the boundary conditions, by utilizing a conservative discretization of the diffusive and, more importantly, the nonlinear migrative flux term. Several numerical experiments are conducted which illustrate that the presented method is first-order accurate in time and second-order accurate in space. Moreover, these tests explicitly indicate that the algorithm is also conservative. Finally we illustrate the applicability of our method in the study of the charging dynamics of porous supercapacitors.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

At the continuum level, aqueous electrolytes (e.g. salty water) can be described as conducting liquids. In such a description, the solvent phase is treated as a standard continuum and discrete ions are replaced by continuous concentration fields. Mutual ion–ion interactions, whether steric or electrostatic, are replaced by interactions between ions and a mean-field potential. The simplest of such models is described by the so-called Poisson–Nernst–Planck (PNP) equations [62,27]. The two most important assumptions used in deriving the PNP equations are:

1. Ions are effectively point size and thus all steric interactions, due to finite ion size, are absent;
2. Ions only interact via a mean-field electrostatic potential field.

Although these assumptions are violated in certain cases [28,7], PNP equations are still very useful in studying many electrochemical and biological phenomena. Examples include, but are not limited to, colloid and interface sciences [62,27],

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

http://dx.doi.org/10.1016/j.jcp.2014.06.039
0021-9991/© 2014 Elsevier Inc. All rights reserved.
electro-osmosis in micro- and nano-fluidic systems [58,59], electrophoretic motion of charged particles [49,27], computation of solvation energies of biomolecules [37], electro-kinetics of electro-chemical cells [8] and the study of supercapacitors [19].

Although our algorithm is quite general and can be applied, at least partially, to many of the application mentioned above, we will focus on the study of supercapacitor. Supercapacitors are energy storage devices that store electrostatic energy inside a very thin, $O(10 \text{ nm})$, layer, called Electric Double Layer (EDL) [27], which forms at the electrode–electrolyte interface (see Fig. 1). To greatly increase the storage capacity, often porous carbon, or other porous conducting materials, are utilized as the electrode. Here, we do not go into the details of these devices and their applications and refer the interested reader to appropriate references [29,57,39].

In designing supercapacitors, it is important to know the effect of the pore micro-structure; in particular its effect on the charging capacity and the charging time of the device. First theoretical studies of supercapacitors date back to the Transmission-Line (TL) model [21]. The TL model is effectively a one-dimensional, equivalent RC circuit model for individual pores in a porous electrode where each pore is represented as a perfect cylinder. Since its introduction, this linear model has been extensively used in studying [52,29,26] and optimizing supercapacitors [23,48]. Despite its widespread use, the TL model faces serious shortcomings. First, the TL theory is only valid at very small voltages ($< 25 \text{ mV}$). In real systems however, applied voltages are typically around 1–2.5 V, depending on the electrolyte used [29]. Second, the TL model assumes that the local EDL thickness is much smaller than the pore radius. This condition, although satisfied for wider macro-pores, does not hold at the nano-pores where pore sizes are on the order of, or even smaller than, the Debye layer [57]. Finally, the actual pore micro-structure are explicitly ignored in TL model.

It is possible to address some of the shortcomings of the TL model, e.g. the low voltage applicability and, to some extent, the effects of micro-structure geometry, by utilizing more sophisticated volume-averaging techniques as initially proposed by [47], and later further developed by Bazant and Biesheuvel [11]. Specifically the model proposed in [11], which we shall refer to the BB model from now on, has been recently used to study various aspects of supercapacitors and similar technologies [53,67,61]. Although the BB model is generally a more realistic model than the TL model, it is nonetheless a volume-averaged model and thus has certain shortcomings. For instance, the averaging implicitness assumes a fairly isotropic porous structure; any anisotropy could lead to uneven transport in different directions. Moreover, like the TL model, the BB model is essentially built on the assumption that the EDL is thin compared to a “reference pore dimension”. Finally, effects of surface conduction phenomena [27] are completely ignored, assuming they are negligible [11].

Despite their popularity, and known shortcomings, very few attempts have been made to directly validate the aforementioned models against a full scale direct numerical simulation (DNS) of the PNP equations at the pore scale. Even the few existing studies are conducted at low voltages; either completely [55] or implicitly for the most part [33]. This is likely due to the numerical challenges associated with the different length scales inherent in the PNP equations at high voltages. In particular, concentration fields often decay exponentially with the local electric potential, which itself decays exponentially with the distance from wall, with a length scale proportional to the EDL thickness [62]. Any successful algorithm must therefore be sophisticated enough to handle such steep gradients efficiently. The PNP equations also involve nonlinear terms describing the flux of ions generated by electric fields. Finally, a useful algorithm, needs to be able to handle the arbitrary and complicated geometry of pore micro-structures (see Fig. 1).

In terms of algorithm development for solving the PNP equations, the biophysics community has been very active. For instance, one-dimensional finite difference methods were utilized by many authors for simulating ion-selective membranes and cellular ion-channels [18,12,32,24,16]. Two- and three-dimension algorithms have also been reported in the literature. In [30,15] authors developed a finite difference, SOR-like algorithm for the steady-state PNP equations. Transient finite difference [68] and finite element [37] methods have also been developed with biophysical applications in mind; though they are general enough that it should be relatively straightforward to apply them to other problems as well. Other communities,
such as electro-chemistry and fluid dynamics, specially since the advent of micro- and nano-fluidic devices, have also been active in this field. For instance, simple one-dimensional finite difference methods have been proposed for studying different aspects of electrochemical cells [46,38,8]. Two- and three-dimensional methods have also been proposed, either for, more or less, general-purpose calculations [31,5] or specific to a particular problem, such as copper electro-deposition [13,14], concentration polarization shocks [66], or electro-kinetic instability [34,22]. For a broader overview of proposed algorithms, we refer the interested reader to a recent review article [63].

Given the existing rich literature, the most distinct feature of this work is the use of adaptive quadtree grids to efficiently handle large gradients in the EDL. Our work is based on successful application of quadtree grids for solving the Poisson–Boltzmann equations [44,45]. However, as discussed in Section 3, this extension requires a novel hybrid approach to ensure that the resulting algorithm is conservative and spurious numerical fluxes do not adversely affect the accuracy of computed solution.

Our paper is organized as follows: Section 2 describes the PNP equations and the relevant boundary conditions. Standard nondimensionalization demonstrates the dependence on two nondimensional variables, namely the electrode potential and the nondimensional EDL thickness. In Section 3.1, we describe how the physical boundary (e.g. of an electrode) is handled via the level-set framework and propose several strategies to automatically generate quadtree adaptive grids that can efficiently resolve large gradients and fluxes in the EDL. In Sections 3.2 and 3.3, we describe the temporal and spatial discretizations in two spatial dimensions. In Section 4, we present several numerical experiments that demonstrate the second-order accuracy in space and first-order accuracy in time. Moreover we explicitly demonstrate the conservativeness properties of our method. Finally, we finish the paper by utilizing our solver to provide DNS evidence supporting the TL model for several pore geometries at low applied voltages. We note that more physically-relevant results, and test of BB model, are delegated to a separate paper [43] which is currently under review.

2. Poisson–Nerst–Planck theory

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

Conservation of ions then requires that [62,27]:

\[
\frac{\partial}{\partial t} c_i = -\nabla \cdot \mathbf{J}_i,
\]

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

\[
\mathbf{J}_i = -D_i \nabla c_i - b_i z_i e \nabla \psi c_i.
\]

In this equation, the first term represents the familiar diffusive flux and \( D_i \) is the diffusion coefficient. The second term is the electro-migrative flux that accounts for the motion of ions due to electrostatic interactions between the electric field, \( \mathbf{E} = -\nabla \psi \), and the electric charge on the ion, \( z_i e \), where \( z_i \) is the ion valency and \( e \) is the fundamental charge of a proton. Finally \( b_i \) is the mobility coefficient that relates the drift velocity of ions to the electric force exerted on them. Mobility coefficient is usually related to diffusion coefficient via Einstein’s relation \( b_i = D_i / k_B T \) where \( k_B \) and \( T \) are the Boltzmann constant and absolute temperature [20], respectively. The mean-field potential, \( \psi \), is related to the ion concentration via the Poisson’s equation:

\[
\nabla \cdot (\varepsilon \nabla \psi) = -\sum_i z_i c_i e,
\]

where \( \varepsilon \) is the permittivity coefficient of the electrolyte.

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

\[
t^* = \frac{t D}{L^2}, \quad \mathbf{x}^* = \frac{\mathbf{x}}{L}, \quad c_i^* = \frac{c_i}{c_\infty}, \quad \psi^* = \frac{\psi e z}{k_B T},
\]

where \( L \) and \( c_\infty \) are reference length and concentration scales, respectively. Rewriting the PNP equations in terms of nondimensionalized variables, after dropping all star superscripts, yields:

\[
\begin{align*}
\frac{\partial}{\partial t} c_+ &= \nabla^2 c_+ + \nabla \cdot (c_+ \nabla \psi) \\
\frac{\partial}{\partial t} c_- &= \nabla^2 c_- - \nabla \cdot (c_- \nabla \psi) \\
\nabla^2 \psi &= -\kappa^2 (c_+ - c_-)
\end{align*}
\]

where \( \kappa = L / \lambda_D \) is the ratio of the reference length scale to the Debye layer thickness; the Debye layer thickness is defined as [54]:
\[ \lambda_D = \sqrt{\frac{e k_B T}{2c_e e^2 z^2}}, \]

and represents the natural length scale over which electrostatic charges on the wall are screened by the ions in the electrolyte and is typically of the order of few nanometers (see Fig. 1).

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

\[ \hat{n} \cdot \nabla c_+ + \hat{n} \cdot \nabla \psi c_+ = 0 \]
\[ \hat{n} \cdot \nabla c_- - \hat{n} \cdot \nabla \psi c_- = 0 \]  \hspace{1cm} (2.4)

(2.5)

However, it should be noted that it is possible to account for reactions through, for example, using the Butler–Volmer reaction kinetics [6]. The boundary conditions for the electrostatic potential is however not unique and greatly depends on the problem under investigation. Usually, one ends up either specifying the potential or the charge density on the boundary, which corresponds to Dirichlet or Neumann boundary conditions, respectively. In general, though, it is possible to use more complex boundary conditions that would, for example, account for effects of dielectric coating or a stern layer [8]. Here we are only concerned with the simple case of a Dirichlet boundary condition for the electric potential, i.e., we impose

\[ \psi = v(x), \] \hspace{1cm} (2.6)

on the electrode’s boundary, where \( v(x) \) is a known function.

3. Numerical approach

3.1. Grid generation and domain description

We choose an implicit surface representation (level-set) to define the electrode’s boundary to ease the treatment of boundary conditions on arbitrary surfaces. This also greatly simplifies the task of generating adaptive grids. To briefly describe the approach in two spatial dimensions, consider the computational domain \( \Omega \) along with its exterior boundary, \( \partial \Omega \), which is subdivided into two disjoint subdomains \( \Omega^+ \) and \( \Omega^- \) by a one-dimensional interface \( I^* \). A level-set function \( \phi \) is used to represent the geometry as:

\[ \begin{cases} 
\phi > 0 & \text{in } \Omega^+ \\
\phi = 0 & \text{on } I^* \\
\phi < 0 & \text{in } \Omega^-.
\end{cases} \]

It is usually desirable that the level-set function be a signed distance function to the interface, i.e., \(|\nabla \phi| = 1\), which may be obtained through the process of reinitialization even if the initial approximation to the level-set function does not satisfy this condition. This subject is very well presented in many resources and we refer the interested reader to appropriate references [50,56,41].

The PNP equations have two length scales that can be identified very easily, namely the EDL scale and the bulk scale. The EDL roughly extends \( O(\kappa^{-1}) \) away from the surface and is characterized by very large gradients in ion concentration and electric potential, whereas these gradients are very small in the bulk. A classical example showing the separation of the two scales is the potential and ion distribution above an infinite planar electrode. At steady state, these are given by the Gouy–Chapman solution [62]:

\[ \psi = 4 \tanh^{-1} \left( \frac{\tanh(v)}{4} \exp(-\kappa x) \right), \] \hspace{1cm} (3.1)

\[ c_\pm = \frac{1}{2} \exp(\mp \psi), \] \hspace{1cm} (3.2)

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

To create adaptive grids, the domain is discretized into cubes (in three spatial dimensions) or rectangles (in two spatial dimensions) that are represented on octree or quadtree data structures, respectively. Fig. 2 illustrates a two-dimensional computational domain along with its corresponding quadtree. Grid generation starts by appointing the tree root, i.e., level zero, to the whole domain and recursively splitting cells of level \( j \) into four (in quadtrees) or eight (in octrees) cells corresponding to level \( j + 1 \). This process is continued until either a certain resolution criterion is met or the tree has reached its maximum level. A tree is called graded if the level difference between any two adjacent cells is at most one and non-graded if there is no such restriction. In this article the more general non-graded case is considered. Finally, a node in...
the tree is called uniform if it is directly connected to other nodes in all directions. Alternatively a node has a T-junction if any of the direct neighbors are missing.

Many different refinement criteria may be defined, with the simplest method being to refine according to the distance from the boundary. Since the level-set function is taken to be a signed distance function, the implementation of this refinement is straightforward: a cell \( C \) with vertices in the set \( V \) is refined if:

\[
\min_{v \in V} |\phi(v)| < \frac{LD}{2},
\]

where \( L \) is the Lipschitz constant of level-set function \( \phi \) and \( D \) is the diagonal size of the current cell [42,60]. This refinement strategy is justified for PNP calculations since the EDL forms close to the interface. Moreover, the EDL thickness, \( \kappa^{-1} \), can be used as a metric to tune the maximum and minimum grid resolutions close to the boundary. We note, however, that this kind of refinement may not be very efficient or enough to capture all the details inside the EDL. A more conservative approach could be refinement based on the gradients in the electric potential and the concentration fields, either using Eqs. (3.1) and (3.2) as rough estimates or via solving the Poisson–Boltzmann equation for the given geometry. We note that these approaches are simply based on physical arguments about the EDL which may not necessarily result in smaller error. This could happen, for example, if the T-junction nodes are present where large gradients exist inside the EDL. In such cases, it may be beneficial to extend the layer of finest grid cells far enough to encompass large gradients.

Refinement could also be done based on trying to minimize or bound the error in the domain. This could be as simple as using the Gouy–Chapman solutions, (3.1) and (3.2), or by utilizing Richardson’s error estimators for the Poisson–Boltzmann equation. Both of these approaches are based on bounding the error at equilibrium. For applications considered in this article, this is a conservative approach since largest gradients only develop toward the end of simulation and at steady state when the EDL is fully charged. However, in other applications with more transient or non-equilibrium nature, such error estimation may not necessarily be accurate. It is possible to apply Richardson error estimators to the transient PNP equations as well. This would truly adapt the grid at each time step but comes at the prohibitive cost of error estimation at every single step. Thus it is not clear if this approach would be any more efficient than the static error estimation.

As can be seen, many different approaches may be taken towards defining the actual refinement criteria. The full analysis of these different refinement criteria is beyond the scope of this paper. As such, here we simply resort to the simplest criteria described by Eq. (3.3) and postpone such an analysis for a future study.

3.2. Time integration

Implicit–Explicit (IMEX) methods [3,2] are quite popular and provide an effective treatment of time discretizations to problems that have both linear and nonlinear parts. The basic idea behind IMEX methods is to treat linear parts of the problem implicitly, hereby avoiding a stringent step restriction, while treating nonlinear parts explicitly. This is specially useful when the linear part comes from the discretization of diffusion operator and is stiff. In the context of the PNP calculations this is even more important since the existence of the EDL imposes a very small grid resolution, \( h_{\text{min}} < \kappa^{-1} \), near the boundary. Moreover, treating nonlinear parts explicitly eliminates the need for linearization and thus avoid the costs associated with computing Jacobians. A general IMEX discretization for the concentration fields in PNP equations is given by:
\[
\sum_{i=0}^{M} \alpha_i c_i^{n+1-i} = -\nabla \cdot \mathbf{j}_c^{n+1} = \nabla^2 c_i^{n+1} - \nabla \cdot \tilde{\mathbf{j}}_c^{n+1}.
\]

Here, the \(\alpha_i\)'s are the discretization coefficients and \(M\) is the degree of discretization in time. Moreover, \(\tilde{\mathbf{j}}_c^{n+1}\) is a consistent approximation to the nonlinear electro-migrative flux at time-step \(t^{n+1}\), i.e. it is required that \(\tilde{\mathbf{j}}_c^{n+1} = \mathbf{j}_c^{n+1} + O(\Delta t^M)\). Extrapolation is utilized to satisfy this condition, i.e. we have

\[
\mathbf{j}_c^{n+1} = \sum_{i=0}^{M-1} \beta_i \mathbf{j}_{c,i}^{n-i},
\]

where the \(\beta_i\)'s are extrapolation coefficients.

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

\[
\frac{c_i^{n+1} - c_i^n}{\Delta t} = \nabla^2 c_i^{n+1} + \nabla \cdot (c_i^n \nabla \psi^n),
\]

(3.4)

\[
\frac{c_i^{n+1} - c_i^n}{\Delta t} = \nabla^2 c_i^{n+1} - \nabla \cdot (c_i^n \nabla \psi^n).
\]

(3.5)

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

\[
-\nabla^2 \psi^{n+1} = \kappa^2 (c_i^{n+1} - c_i^{n-1}).
\]

(3.6)

3.3. Spatial discretization

Physically, the Poisson equation (2.3) is very different from the Nernst–Planck equations (2.2) and (2.3). Specifically, the Nernst–Planck equations describe conservation laws for ion concentrations whereas the Poisson equation, in the context of PNP equations, does not have this property. As a result it is desirable, and at large voltages necessary, that the discrete form of the Nernst–Planck equations also conserves the mass both locally and globally.

The nodal discretizations on quadtree adaptive grids of [42,17,25] do not satisfy conservation properties at T-junctions but produce second-order accurate solutions and second-order accurate approximations of gradients. As a result nodal approximations are appropriate, and desirable, for the spatial discretization of the Poisson equation. We will therefore utilize the same nodal approach for the Poisson equation. Since the discretizations along with the treatment of boundary condition is very similar to the Poisson–Boltzmann equation [44,45], we simply refer the interested reader to the aforementioned articles and the references therein.

As mentioned earlier, large gradients exist inside the EDL, especially at large voltages. Since the Nernst–Planck equations describe conservation properties, it is desired to utilize conservative discretizations when possible. This is specially important since T-junctions generally exist inside EDL where gradients are large and non-conservative properties of nodal discretizations are amplified. Various conservative methods on adaptive Cartesian grids have been proposed, most notably for discretization of Hyperbolic Conservation Laws [10,9] and Navier–Stokes equations [1,36,35]. Here, the method described in [35] is chosen for the discretization of the diffusion part whereas the electro-migrative part is discretized using a novel hybrid approach, detailed in Section 3.3.2.

3.3.1. Diffusion

Since the discretizations of Eqs. (3.4) and (3.5) are similar, we only consider the discretization of the equation for the cations and drop the \(+\) sign for brevity. Let us consider the concentration variable, \(c\), that is stored at the center of cell \(C_c\) as illustrated in Fig. 3. Integration of the left-hand side of (3.4) is trivial and yields:

\[
\int_{C_c} \frac{c_i^{n+1} - c_i^n}{\Delta t} \, dA \approx \frac{c_i^{n+1} - c_i^n}{\Delta t} A_c,
\]

(3.7)

where \(A_c\) is the area of cell \(C_c\) which is simply \(A_c = \Delta x_c \Delta y_c\) if the cell is entirely in \(\Omega^+\). If the cell \(C_c\) is crossed by the boundary \(\Gamma\), we utilize the geometric integration technique introduced in [40] to evaluate the cell area. The diffusion part is discretized using the divergence theorem:

\[
\int_{\partial C_c} \nabla c \, dA = \int_{\partial C_c} \mathbf{n} \cdot \nabla c \, d\ell.
\]

Consider the discretization of the flux term at the right boundary, \(x^+\), of a general cell configuration as depicted in Fig. 3(a). Following [35], we write:
where \( \bar{F}^d \) is the average diffusive flux density across all the cells at the right boundary and is written as:

\[
\bar{F}^d = \frac{1}{\Delta y_c} \sum_{j=1}^{N_{xc}} F^d_j \Delta y_j,
\]

\[
= \frac{1}{\Delta y_c} \sum_{j=1}^{N_{xc}} c_j - c_c \Delta y_j.
\]

Here, \( \bar{\Delta x} \) is the average distance between cell centers and is written as:

\[
\bar{\Delta x} = \frac{1}{\Delta y_c} \sum_{j=1}^{N_{xc}} \frac{\Delta x_c + \Delta x_j}{2} \Delta y_j.
\]

Notice that the definition of \( \bar{F}^d \) and \( \bar{\Delta x} \) are the same for all the cells that share the right boundary, i.e. the same calculated values are used for both \( c_c \) and all the \( c_j \)'s alike. This leads to a conservative symmetric discretization of the diffusion operator by combining Eqs. (3.8) through (3.11):

\[
\int_{x^+}^{x} \mathbf{n} \cdot \nabla c \, d\ell \approx \frac{2}{\sum_{j=1}^{N_{xc}} (\Delta x_c + \Delta x_j) \Delta y_j} \sum_{j=1}^{N_{xc}} \Delta y_c \Delta y_j (c_j - c_c).
\]

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

### 3.3.2. Electro-migration

Computation of the electro-migration term involves both the concentration, stored at cell centers, and the potential, stored at cell vertices. Starting from the differential form (3.4) and applying the divergence theorem, we obtain:

\[
\int_{c_c} \nabla \cdot (c \nabla \psi) \, dA = \int_{\partial c_c} \mathbf{n} \cdot \nabla \psi c \, d\ell.
\]

Again, let us consider the discretization of the flux term at the right boundary, \( x^+ \) (see Fig. 3(b)). Similar to Eq. (3.8), we write:

\[
\int_{\partial c_c} \mathbf{n} \cdot \nabla \psi c \, d\ell \approx \bar{F}^e \Delta y_c.
\]

Here, \( \bar{F}^e \) is the average electro-migrative flux, where the averaging is defined as in (3.9) and individual fluxes are written as:

\[
F^e_j = \frac{1}{2} (\bar{\psi}_x|v_j + \bar{\psi}_x|v_{j+1}) c_{j+1/2},
\]

where \( \bar{\psi}_x \) is the second-order nodal approximation to \( \partial \psi/\partial x \) as described in [42,17,25], \( c_{j+1/2} \) is a second-order accurate approximation of the concentration field evaluated at \( (x_c + \Delta x_c/2, y_j) \). The simplest approach for obtaining such a second-order accurate result involves linear interpolation of concentration values, which can be done by considering a triangulation.
of cell centers. Moreover, such an interpolation scheme is also necessary when solving the Poisson equation. We therefore postpone the full description of such interpolation scheme to Section 3.3.3. Combining (3.13) and (3.14), we obtain the final form of the discretization:

$$\int_{\partial C_i} \hat{n} \cdot \nabla \psi c \, d\ell \approx \frac{1}{2} \sum_{j=1}^{N_x} \left( \sum_{k=1}^{3} \psi_{c_{j,k}} \mathbf{x}_{v_j} \right) \Delta y_j.$$  

(3.15)

Just as before, application of discretization (3.15) to remaining cell boundaries is trivial. We also note that the extension to three spatial dimensions, though technically more involved, follows the same logic as presented here. One should note, however, in three spatial dimensions, cell data must be interpolated on tetrahedrons instead of triangles.

3.3.3. Cell data interpolation

As discussed earlier, a second-order accurate, cell-based interpolation scheme is required in our method. This is needed when in computing $c_{j+1/2}$ in (3.14) and when evaluating the source term in the Poisson equation (3.6) at the nodes. Unlike node-based interpolations, cell-based interpolations are expensive to construct on quadtree. This is because cell centers, unlike nodes, do not follow a particular logical coordination and generally a triangulation is necessary. As a result we use linear interpolation on a locally triangulated grid as illustrated in Fig. 4.

To construct the local triangulation, we merely generate a list of all neighboring cell centers in an arbitrary order (clockwise or counter-clockwise). Note that no effort is made to obtain “optimum” triangulation in any sense. Given an arbitrary point $\mathbf{x}$ in the triangulation $K = \bigcup_{j=1}^{N} K_j$, where $K_j$’s are triangles, it is first required to find the triangle $K_j(p_0, p_j, p_j+1)$ such that $\mathbf{x} \in K_j$. This can easily be done by mapping the triangle $K_j$ to the reference triangle in the $(u, v)$ domain (see Fig. 4). The mapping is given by:

$$\begin{pmatrix} x_l - x_0 \\ x_{l+1} - x_0 \\ y_l - y_0 \\ y_{l+1} - y_0 \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix},$$

which results in:

$$u = \frac{\mathbf{r} \times \mathbf{r}_{j+1}}{\mathbf{r}_j \times \mathbf{r}_{j+1}}, \quad v = \frac{\mathbf{r} \times \mathbf{r}_j}{\mathbf{r}_j \times \mathbf{r}_{j+1}}.$$  

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

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

$$c(\mathbf{x}) = \sum_{l=0}^{2} c_l N_l(u, v),$$

where $c_0 = c_0$, $c_1 = c_j$, and $c_2 = c_{j+1}$.

3.3.4. Boundary conditions

To impose boundary conditions we use the cut-cell method [51,25]. Consider cell $C_i$ cut by the boundary $\Gamma$ as illustrated in Fig. 5. We note that our grid generation scheme, based on the level-set method, ensures that the grid is locally uniform close to the boundary, which facilitates the discretization of boundary conditions. To impose the boundary conditions, let us consider only the cations and note that Eq. (2.5) may be written as:
\[ \hat{n} \cdot J = \hat{n} \cdot (J_L + J_e) = 0, \]

which includes both the diffusion and the electro-migration terms. It is then concluded that the total flux term \( J^{n+1} \) disappears on the electrode boundary, i.e. \( \hat{n} \cdot J^{n+1} \big|_{r} = 0 \). Thus, integrating the right-hand side of Eq. (3.4) in \( C_c \cap \Omega^- \) yields:

\[
\int_{C_c \cap \Omega^-} \nabla^2 \varphi^{n+1} + \nabla \cdot (c^n \nabla \psi^n) \, dA = \int_{\partial C_c \cap \Omega^-} \hat{n} \cdot (\nabla \varphi^{n+1} + c^n \nabla \psi^n) \, dl,
\]

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

\[
\int_{x^+ \cap \Omega^-} \hat{n} \cdot \nabla \varphi^{n+1} \, dl = c_j^{n+1} - c_c^{n+1} \frac{\Delta x}{\ell_{x^+}}, \tag{3.16}
\]

where \( 0 \leq \ell_{x^+} \leq \Delta y \) is the length of right side of cell \( C_c \) that is in \( \Omega^- \) and is computed as:

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

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

\[
\int_{x^+ \cap \Omega^-} \hat{n} \cdot \nabla \psi^n c^n \, dl = \frac{\bar{\psi}_{x^+}^{++} + \bar{\psi}_{x^+}^{--}}{2} c_c^n + \frac{c_j^n}{2} \ell_{x^+}. \tag{3.17}
\]

### 3.3.5. Extrapolation

Accurate calculation of gradients, right at the boundary, requires data that resides in the \( \Omega^+ \) domain, which may not be available since the solution is computed in the \( \Omega^- \) domain. As a result it is necessary to be able to “extend” valid solutions from \( \Omega^- \) into \( \Omega^+ \). This can be achieved via extrapolation methods. In [4], authors proposed an extrapolation method based on solving a series of auxiliary PDEs, which is robust and can easily be extended to achieve arbitrary polynomial order. While we have had success with this approach in the context of Poisson–Boltzmann solvers [44,45], the proposed method is computationally expensive. We therefore propose an alternative, geometric approach to the extrapolation problem that is significantly faster. This methodology may not be as robust if the interface is under-resolved and therefore one must impose that the interface is well resolved at all times.

Our new method is based on building and extrapolating polynomials in the normal direction to the interface. Fig. 6 schematically depicts a situation where a quantity of interest (the potential field for example) needs to be extrapolated from \( \Omega^- \) into \( \Omega^+ \). The first step in constructing the polynomial involves projecting the point \( x_i \) onto the interface. The location of the projection point \( x_0 \) can easily be calculated from the level-set function if it is a signed-distance function:

\[
x_0 = x_i - \phi \hat{n}, \tag{3.18}
\]

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

\[ F_0 = F_0 = F_D(x_0), \]

where it is assumed that a Dirichlet boundary condition, \( F_D \), is available for the quantity \( F \). If such an information is not available or a higher order method is desired, more points \( x_i \) in the normal direction may be obtained according to
Eq. (3.18). Next, and depending on the desired order of accuracy, further points in the normal directions are computed according to (3.19). Note, however, that in regions of the domain where curvature of the boundary is larger than what can be captured by the grid, it is necessary to lower the degree of extrapolating function. This is done by removing points that will result in use of invalid data from \( \Omega^- \) domain. Here green squares represent points in the \( \Omega^+ \) domain for which three valid points (including the projection point) may be computed which results in a quadratic extrapolation function. This is in contrast to yellow and red squares for which only two and one points are found, respectively. In these cases a linear and constant polynomials are used for extrapolation, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

\[
\mathbf{x}_j = \mathbf{x}_0 - \delta_j \mathbf{n}, \quad j = 1, 2, \ldots \tag{3.19}
\]

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

The simplest approach to chose the points \( \mathbf{x}_j \) that satisfies all three conditions is to choose them equidistant to each other, i.e. \( \delta_j = j\sqrt{\Delta x^2 + \Delta y^2} \). Furthermore, for such a point, \( F_j \) can simply be evaluated using a bilinear or biquadratic interpolating function \( F_1(\mathbf{x}) \) [41]. One should note that these interpolating functions may not always be available. For instance if the interface is not smooth enough or accurately resolved by the grid, it is possible that one or more \( \mathbf{x}_j \) points end up too close to the interface, or even cross the interface again and end up in \( \Omega^+ \) (see Fig. 6). In both situations, the interpolating function \( F_1 \) could potentially include points that belong to \( \Omega^- \) and have invalid values.

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

1. Compute the projected point \( \mathbf{x}_0 \) using Eq. (3.18).
2. If a Dirichlet boundary condition, \( F_D \), is available for \( F \), compute \( F_0 = F_D(\mathbf{x}_0) \) and add \( (\mathbf{x}_0, F_D) \) to the polynomial list. Otherwise, skip this step.
3. For \( j = 1 : M \), compute the point \( \mathbf{x}_j \) according to Eq. (3.19). If \( \mathbf{x}_j \) can be used for the extrapolation procedure, add \( (\mathbf{x}_j, F_j(\mathbf{x}_j)) \) to the polynomial list. If it cannot be used for the extrapolation procedure, terminate the loop. Here we use a biquadratic interpolating function for \( F_j \) for increased accuracy. However, bilinear methods are preferred if the interface is not very smooth or is under-resolved.
4. Construct the extrapolating polynomial \( F_E \) using as many points as possible using Newton polynomials:

\[
F_E(\eta) = \sum_{j=0}^{M} [F_0, F_1, \ldots, F_j] N_j(\eta),
\]

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

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

5. The extrapolated value at \( \mathbf{x}_i \) simply becomes \( F_i = F_E(\phi_i) \).

3.4 Notes on extension to three spatial dimensions

The algorithm presented in this work was primarily targeted toward two-dimensional problems. However, most parts of the algorithm can easily be extended to three spatial dimensions. Specifically, handling the Poisson equation (3.6) in
three spatial dimensions is analogous to treatment of the Poisson–Boltzmann equation which is explained in detail in [44] and references therein. Also, the finite volume discretization of diffusion and electro-migration parts, as presented in Sections 3.3.1 and 3.3.2, conceptually remain unchanged. The only required modification is to replace surface and line integrals to volume and surface integrals, respectively, and evaluate fluxes at the face center rather than edge center (see [35] for similar approaches). For cells that are far away from the interface this modification is trivial. For cells that are cut by the interface, however, one is required to replace the length fraction, \( l_{x^+} \), with an appropriate face fraction term. The computation of this face fraction, along with any possible integration on the interface, can be achieved using the geometric integration technique introduced in [40].

One ingredient of our algorithm that does not trivially extend to three spatial dimensions in the cell-based interpolation discussed in Section 3.3.3. Note that although the general approach, i.e. linear interpolation based on triangulation of cell centers, is still viable, the way this triangulation is built in two dimensions does not extend to three dimensions. One way to solve this problem is to use a Delaunay triangulation algorithm that only involves the neighboring cell centers. Another way to approach this problem, which does not involve triangulation at all, is to construct a local, linear, least-squares interpolation that only involve neighboring cells. This is always possible since each cell has at least one neighboring cell in each of \( x, y, \) and \( z \) directions. Finally, it is easy to note that the extrapolation technique introduced in Section 3.3.5 trivially extends to three dimensions provided that a cell-based interpolation is available.

4. Accuracy analysis

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

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

\[
f_h = f + C h^r + o(h^r),
\]

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

\[
f_{h/2} = f + C \frac{h^r}{2^r} + o(h^r), \quad f_{h/4} = f + C \frac{h^r}{4^r} + o(h^r).
\]

As a result, the convergence rate can be estimated as

\[
r \approx \log_2 \left( \frac{\| f_h - f_{h/2} \|_p}{\| f_{h/2} - f_{h/4} \|_p} \right), \tag{4.1}
\]

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

\[
\| e_h \|_p = \| f_h - f \|_p \approx \left\| \frac{2^r}{2^r - 1} (f_h - f_{h/2}) \right\|_p. \tag{4.2}
\]

4.1. Conservativeness

4.1.1. Two-dimensional cavity

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

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

Finally, a grid resolution of \( l_{\text{max}} = 8, l_{\text{min}} = 4 \) and a constant time-step of \( \Delta t = 5 \times 10^{-3} \) is chosen for this calculation. Fig. 7(b) illustrates a snapshot of the charge density distribution at final time along with the adaptive grid in the domain.
To measure conservativeness, the simulation is initiated at $t = 0$, with uniform concentration fields $c_{\pm} = 0.5$, and integrated up to $t = 1$. At each time, the total mass in the system is calculated via:

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

and the relative change in the mass is used as a metric for the conservativeness of the method. Since this is a closed system, the total mass in the system should not change in time. Indeed Fig. 7(a) illustrates that the method does conserve mass.

4.1.2. Concentric circular electrodes

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

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

$$\phi_1(x, y) = 0.25 - \sqrt{x^2 + y^2},$$

$$\phi_2(x, y) = \sqrt{x^2 + y^2} - 0.75,$$

which defines $\Omega^-$ as the space between two concentric circles. All of the simulation constants are kept the same as their values in the previous test. The only exception is for the potential which is kept at 5 and $-5$ thermal volts on the inner and outer circles, respectively. Fig. 8(a) illustrates that the method remains conservative even in the presence of complex geometry. Fig. 8(b) illustrates a snapshot of the solution along with the adaptive grid at final time.

4.2. Spatial convergence rate

In this section we consider a series of different geometries and solve the PNP equations using our algorithm. In each case, a base grid is chosen and during successive refinement, every cell is split to ensure uniform convergence in the domain. Moreover, to ensure that the error due to time discretization is also reduced at the same rate, time-step is reduced such that $\Delta t \propto h_{\text{min}}^{2}$ at each refinement step. Note that this is done to ensure uniform decay of temporal errors and should not be confused with a stability criterion (cf. Section 4.3).

On each grid the PNP equations are integrated up to a final time after which Eqs. (4.1) and (4.2) are utilized to measure the convergence rate and error in space. Since the calculation of each convergence rate in (4.1) requires 3 consecutive refinements, a total of 5 refinement steps are chosen. Once consecutive error terms are computed using (4.2), the overall convergence rates are computed via fitting the following curve to the error:

$$\|e_h\|_p = Ch_{\text{min}}^r.$$  \hspace{1cm} (4.3)

Note that here we are mainly concerned with the convergence in space. Further discussion of choosing a proper time-step and convergence in time is provided in Section 4.3.
The convergence of the solution determined by the \( \Delta t \) is due to using iterative linear solver for the diffusion part and round-off errors.

\[ \phi(x, y) = \sqrt{x^2 + y^2}. \]

The potential is set to \(-3\) thermal volts on the boundary and zero at the box boundaries of the domain, \( \partial \Omega \). Ion concentrations satisfy the no-flux boundary condition on the electrode and the “bulk” condition at the box boundary, i.e. \( c_\pm(x, y) = 0.5 \) for \((x, y) \in \partial \Omega\). Finally an EDL thickness of \( \kappa^{-1} = 0.03 \) is chosen for this problem. Note that these numbers are chosen simply because they are physically relevant in studying supercapacitors. A base grid of \((l_{\text{max}} = 8, l_{\text{min}} = 4)\) along with a coarse time-step of \( \Delta t_c = 5 \times 10^{-3} \) is chosen for grid refinement. Finally, periodic boundary conditions are chosen for all quantities in the \( x \)-direction. Fig. 9 illustrates both the base grid and the solution at the final time of \( t_f = 100 \times \Delta t_c = 0.05 \) on the finest grid. Fig. 10 gives the convergence rate in the \( L_1, L_2 \) and \( L_\infty \) norms under grid refinement. Notice that the number in parentheses is the overall convergence rate determined by fitting Eq. (4.3).

4.2.1. Flat electrodes

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

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

A base resolution of \((l_{\text{max}} = 8, l_{\text{min}} = 4)\) is chosen for the grid with a coarse time-step of \( \Delta t_c = 5 \times 10^{-3} \). The electric potential on the electrodes is set to \(5\) and \(-5\) thermal volts for the top and bottom walls, respectively. Also a thin EDL limit of \( \kappa^{-1} = 0.1 \) is considered. Finally, periodic boundary conditions are chosen for all quantities in the \( x \)-direction. Fig. 11 illustrates both the base grid and the solution at the final time of \( t_f = 100 \times \Delta t_c = 0.05 \) on the finest grid. Fig. 10 gives the convergence rate in the \( L_1, L_2 \) and \( L_\infty \) norms under grid refinement. Notice that the number in parentheses is the overall convergence rate determined by fitting Eq. (4.3).

4.2.2. Circular electrode

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

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

The potential is set to \(-3\) thermal volts on the boundary and zero at the box boundaries of the domain, \( \partial \Omega \). Ion concentrations satisfy the no-flux boundary condition on the electrode and the “bulk” condition at the box boundary, i.e. \( c_\pm(x, y) = 0.5 \) for \((x, y) \in \partial \Omega\). Finally an EDL thickness of \( \kappa^{-1} = 0.03 \) is chosen for this problem. Note that these numbers are chosen simply because they are physically relevant in studying supercapacitors. A base grid of \((l_{\text{max}} = 8, l_{\text{min}} = 3)\) along with a coarse time-step of \( \Delta t_c = 4.5 \times 10^{-4} \) are chosen. Fig. 11 illustrates the initial grid and level-set along with the final solution on the finest grid while, Fig. 12 demonstrates that our method is second-order in all three norms for this numerical example.
Fig. 10. Convergence rate of our algorithm for 4.2.1, obtained via Richardson extrapolation method. The numbers in parenthesis denote the overall rate obtained via fitting Eq. (4.3).

Fig. 11. Solution of PNP for test 4.2.2 at the final time of $t_f = 100 \times \Delta t_c = 0.045$. The base grid along with the zero level-set is shown in (a). The numerical solutions (b–d) are obtained on the finest grid using the finest time-step. The EDL is marked with a white circle, representing the $\phi = -\kappa^{-1} = -0.03$ contour. The black shaded area represents the electrode ($\Omega^+$ domain).

Fig. 12. Convergence rate for test 4.2.2 using the Richardson extrapolation method. The numbers in parenthesis denote the overall rate obtained via fitting Eq. (4.3).

4.2.3. Star-shaped electrode

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

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

Similar boundary conditions are imposed for the potential and ion concentrations as in the previous test whereas a larger EDL with thickness of $\kappa^{-1} = 0.1$ is chosen. As for numerical parameters, a base grid of $(l_{\max} = 8, l_{\min} = 5)$ along with a coarse time-step of $\Delta t_c = 5 \times 10^{-3}$ are chosen. Fig. 13 illustrates the initial grid and the zero level-set along with the final solutions on the finest grid. Again, Fig. 14 demonstrates that our algorithm is second order in all three norms for this numerical example.
4.2.4. Non-smooth electrode

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

\[
\phi(x, y) = \min(\phi_1, \phi_2),
\]

where \( \phi_1 \) is given by:

\[
\begin{align*}
\phi_1(x, y) &= \min(f_1, f_2), \\
f_1(x, y) &= \sqrt{2|x| + |y|} - 1, \\
f_2(x, y) &= \sqrt{2|y| + |x|} - 1,
\end{align*}
\]

and \( \phi_2 \) is given by a 45 degree rotation about the origin of \( \phi_2' \):

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

The resulting boundary is shown in Fig. 15 with an \( (l_{\text{max}} = 8, l_{\text{min}} = 5) \) adaptive grid. Here we keep all the parameters unchanged from the previous test with the exception of potential on the electrode which is set to \(-1\) thermal volts. The final solution at \( t_f = 100 \times \Delta t_f = 0.05 \) on the finest grid is shown in Fig. 15.

Fig. 16 illustrates the convergence rate analysis of our method. Unfortunately, this boundary contains a singularity which cannot be resolved by grid refinement. As a result, no matter how much the grid is refined, there will always be points close to the singularity for which one has to reduce extrapolating function to a constant polynomial. As a result, convergence rate in \( L_{\infty} \) is dropped by at least one order. Convergence rate in \( L_1 \) and \( L_2 \), however, is not affected by the singularity, meaning that loss of accuracy is purely a local property.

These results are typical of parabolic-type problems with geometrical singularities in the domain. Existing methods have been proposed to overcome this difficulty in the literature. For instance in [65] and [64] authors propose the so-called “Matched Interface and Boundary” (MIB) method for handling geometrical singularities for elliptic problems with applications to the Poisson–Boltzmann equation. In general it is possible to incorporate this method in our algorithm since the
and with in φ(4.3).

For modifications and fitting Fig. 16. Convergence area solutions Fig. 15. Solution of PNP for test 4.2.4 at the final time of \( t_f = 100 \times \Delta t_c = 0.05 \). Base grid along with the zero level-set is shown in (a). The numerical solutions (b–d) belong to the finest grid using the finest time-step. EDL is marked with a white curve, representing \( \phi = -\kappa^{-1} = -0.1 \). The black shaded area represents the electrode (\( \Omega^+ \) domain).

Fig. 16. Convergence rate of test 4.2.4 problem using Richardson extrapolation method. The number in parenthesis denotes the overall rate obtained via fitting Eq. (4.3). Notice how existence of singularities has dropped the convergence rate by at least one order in \( L_\infty \). This, however, is a local loss since \( L_1 \) and \( L_2 \) norms still show second-order convergence.

modifications only affect the boundary conditions where the grid is locally uniform. Nonetheless, for most practical applications, such as studying the charging kinetics of supercapacitors, the geometry does not involve geometrical singularities. For these problems the maximum curvature in the geometry, albeit maybe large, is finite and is eventually resolved on a fine enough grid. Although this may be prohibitively expensive on a uniform grid, our use of quadtree adaptive grids can dramatically reduce the computational cost in these cases.

4.3. Analytic test

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

\[
\psi(x, y, t) = \exp(\kappa \phi(x, y))(1 - \exp(-t)),
\]

\[
c^+(x, y, t) = \frac{1}{2} \exp(-\psi(x, y, t)),
\]

\[
c^-(x, y, t) = \frac{1}{2} \exp(\psi(x, y, t)),
\]

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

\[
\partial_t c_+ = \nabla^2 c_+ + \nabla \cdot (c_+ \nabla \psi) + f(x, y, t),
\]

\[
\partial_t c_- = \nabla^2 c_- - \nabla \cdot (c_- \nabla \psi) + g(x, y, t),
\]

\[-\nabla^2 \psi = \kappa^2 (c_+ - c_-) + h(x, y, t),\]

with the source terms \( f, g, \) and \( h \) obtained by plugging the exact solutions into the PNP equations. Moreover, these exact solutions are also utilized to obtain appropriate Dirichlet and no-flux boundary conditions on the electrode boundary, \( \Gamma \), and domain boundary, \( \partial \Omega \) (see Eqs. (2.5), (2.5), and (2.6)).
To demonstrate the robustness of our algorithm, we initially apply the regular grid generation scheme discussed in Section 3.1 using a base resolution of \((l_{\text{max}} = 7, l_{\text{min}} = 3)\). Next, the tree is randomly refined to ensure that the resulting grid is non-graded. This is illustrated in Fig. 17(a). Finally, the EDL thickness is set to \(\kappa^{-1} = 0.25\) and an initial time-step of \(\Delta t = 3.125 \times 10^{-2}\) is chosen for the integration. Once the errors are calculated, the overall temporal convergence rate, \(s\), is computed by fitting \(\|e_h\|_p = C \Delta t^s\) to the error. As expected, Fig. 18 illustrates first order convergence rate for all variables in \(L_1\), \(L_2\), and \(L_\infty\) norms.

Finally, to study the convergence rate in space, the PNP equations are integrated up to \(t = 10\) using a fixed time-step of \(\Delta t = 3.125 \times 10^{-2}\). Since the solution involves an exponential decay in time, temporal errors are minimal at \(t = 10\) and the main contribution to the error is from spatial discretization. Fig. 19 illustrates that our algorithm is second order accurate in space, even when the underlying grid involves large jumps in the level of neighboring cells. Here, the overall convergence rate in space is calculated by fitting Eq. (4.3) to the error as before.

We finish this section by noting that the presented algorithm is conservative, first order accurate in time and second order accurate in space, even for non-graded grids. An important question that we have left unanswered here regards the stability of our algorithm. While theoretical stability analysis of the algorithm is of great interest, it has proven to be
quite challenging due to nonlinear nature of the equations and grid adaptivity. Nonetheless, in our studies we have never experienced any time-step restrictions with respect to the grid size or the jump in the level of neighboring cells. These observations are consistent with those made in [37].

The only time-step restriction we have encountered so far, seems to solely depend on the EDL thickness, \( \kappa^{-1} \), and the applied potential. In particular, we have found \( \Delta t \leq \kappa^{-2} \) as a reliable upper bound at small potentials. This is due to explicit treatment of the nonlinear electro-migration term and is consistent with the fact that EDL relaxation time scales as \( \tau = \mathcal{O}(\kappa^{-2}) \) [8]. At higher potentials, a smaller time-step is typically required but in this article we are unable to give any specific recommendation.

5. Application to modeling supercapacitors

In this section we show an application of our algorithm in verifying the validity of the Transmission-Line (TL) model [21] via Direct Numerical Simulations (DNS). Since its advent in the 60’s, the TL model has been widely used in studying supercapacitors and other devices that make use of porous conducting electrodes (see Section 1). There are several assumptions that go into the derivation of the TL model. First, it is assumed that the porous material is made of perfectly circular cylinders. Second, it is assumed that the EDL is thin compared to the pore radius, i.e. \( \lambda_D/r \ll 1 \). Finally, it is assumed that the applied voltage is very small compared to the thermal voltage, i.e. \( \Delta V \ll \varepsilon z/k_B T \).

The TL model was later extended to include more general porous structures by means of volume-averaging. This was initially done with battery applications in mind [47] and recently extended to study supercapacitors [11] by utilizing matched asymptotic techniques and assuming Gouy–Chapman solutions for the EDL [8]. We refer to the model proposed in [11] as the 'BB' model for short. Indeed the BB model extends the TL model in two important ways: First, it can be applied to arbitrary pore geometries as long as the porous material is homogeneous to justify volume-averaging. Second, it can be applied to systems where the electrode voltage exceeds the thermal voltage (see Section 2). Just like the TL model, however, the BB model can only be applied to systems where the EDL is thin compared to the macroscopic pore width defined below.

Consider a homogeneous porous electrode schematically shown in Fig. 1. The macroscopic pore width is defined in terms of the total volume and surface area of all pores:

\[
\eta_p = \frac{\int_{\Omega^+} \frac{d\Omega}{r}}{\int_{\Omega^-} \frac{d\Omega}{r}}. 
\]

The BB model is considered to be a valid model if the EDL is thin, i.e. if \( \kappa^{-1} = \lambda_D/h_p \ll 1 \). Of course both the TL and the BB models can be compared against a full scale DNS of the PNP equations to test if they appropriately model supercapacitors in different regimes and to understand their weaknesses. A through study of these models deserves its own paper, which is currently under review [43]. However, to show the applicability our algorithm, we consider studying the simpler case of the TL model. In particular, we are interested in studying the effect of the pore micro-structure on the charging dynamics and the time scales. To be able to compare the DNS results with the TL model, let us define a charging fraction parameter as:

\[
\eta(t) = \frac{q(t)}{q(t \to \infty)}. 
\]

where \( q(t) \) is the total amount of charge that is stored in the electrode at time \( t \):  

\[
q(t) = \int_{\Omega^-} (c^+ - c^-) \, d\Omega. 
\]

The TL model (as derived from the BB model at low voltages), predicts that the non-dimensional average potential field \( \psi(x, \tau) \) evolves according to:

\[
\frac{\partial \psi}{\partial \tau} = \frac{\partial^2 \psi}{\partial x^2}. 
\]

(5.3)

Here, the potential field is non-dimensionalized by the thermal voltage (see Section 2), space is non-dimensionalized by the pore half-length, \( L \) (see Fig. 1), and time is non-dimensionalized using the Transmission-Line charging time given by:

\[
\tau_{TL} = \frac{\lambda_D L^2}{h_p D}. 
\]

(5.4)

Eq. (5.3) can easily be solved analytically subjected to \( \psi(0, \tau) = 0, \psi_x(1, \tau) = 0, \) and \( \psi(x, \tau) = V, \) where \( V \) is the applied voltage on the electrode. The solution to (5.3) can be used to obtain an expression for the charging fraction predicted by the TL model. This is given by:

\[
\eta(t) = 1 - \sum_{n=0}^{\infty} \frac{2}{n^2 \lambda_n^2} \exp(-\lambda_n^2 t), 
\]

(5.5)

where \( \lambda_n = n\pi + \pi/2. \)
Finally, it is important from the practical point of view as it would suggest certain designs could lead to fast charging. However, the TL model (5.5) only predicts a single charging time scale given via Eq. (5.4), which only depends on \( h_p \) and not the actual micro-structure itself. This is beautifully shown to be true if the time variable from the DNS is scaled with the TL time scale in Fig. 20(a) (main plot).

Indeed, what Fig. 20(a) illustrates is a verification of the validity of the TL model using DNS of the PNP equations without any implicit assumptions. Note that, however, this is only the case at low applied voltages. What happens at large voltages is not captured correctly by the TL model as discussed in [43]. We close this section by providing snapshots of the potential field in the pores. To better illustrate the applicability of the TL model, DNS snapshots are chosen such that they correspond to \( t/\tau_{TL} = 10^{-2}, 10^{-1} \) and 1. Note that since each porous structure is associated with a different macroscopic pore width, these snapshots correspond to different actual simulation times. Nonetheless, as the TL suggests, the potential fronts and shapes look very similar in each snapshot across different geometries.

6. Conclusions

In this paper we have presented a conservative semi-implicit algorithm for the solution of the Poisson–Nernst–Planck equations. Our method is unique as it resolves the thin electric double layer without any ad-hoc assumption about bulk electro-neutrality. It is also efficient since it utilizes quadtree data structures to create adaptive Cartesian grids that can easily resolve exponentially varying quantities in the electric double layer. Moreover, our approach to this problem is novel as it combines the conservative properties of finite volume methods with the accuracy and ease of use of finite difference methods on quadtree grids. Finally, a fast geometric extrapolation scheme is presented.
Through numerical examples, we have clearly demonstrated that our method is indeed conservative, in the sense that the total number of ions in a closed system is always preserved. Moreover, several examples have been chosen to demonstrate the second-order accuracy of the method. The numerical examples have been chosen to mimic the essential challenges involved with solving the Poisson–Nernst–Planck equations, i.e. thin electric double layer and large applied voltage limits. Finally, the algorithm was used to validate the transmission-line model of supercapacitors. This is done by comparing the transmission-line solution for a set of different porous electrodes to the direct numerical simulation of the full Poisson–Nernst–Planck equations, without any implicit assumptions about the electric double layer thickness or the pore micro-structure. Further study and validation of other low-order models are deferred to a companion paper that is currently under review [43].

Finally, we would like to note that in this paper we have only described the general algorithm that would work on any adaptive Cartesian grid. However, there are several ways that this algorithm may be extended. Most obviously, we have omitted the discussion of higher order time integration schemes, partly because they are relatively easy to devise but also because they may require stability constraints that need to be studied separately. More importantly, we did not discuss the most effective strategy for generating adaptive quadtree grids for the Poisson–Nernst–Planck equations. This is beyond the scope of the present paper and is postponed to a separate paper for future investigation.

Acknowledgements

We would like to thank Dr. Todd Squires for stimulating discussions about the need for developing a computational framework for the PNP equations. We would also like to thank anonymous reviewers for their invaluable suggestions that have lead to overall improvement of the quality of this article. This work was supported in part by ONR N00014-11-1-0027 and by the W.M. Keck Foundation (grant no. SB110056).

References
