



Electrochemical transport modelling and open-source simulation of pore-scale solid–liquid systems

Robert Barnett¹ · Federico Municchi² · John King¹ · Matteo Icardi¹

Received: 27 December 2022 / Accepted: 19 April 2023 / Published online: 26 September 2023
© The Author(s) 2023

Abstract

The modelling of electrokinetic flows is a critical aspect spanning many industrial applications and research fields. This has introduced great demand in flexible numerical solvers to describe these flows. The underlying phenomena are microscopic, non-linear, and often involving multiple domains. Therefore often model assumptions and several numerical approximations are introduced to simplify the solution. In this work we present a multi-domain multi-species electrokinetic flow model including complex interface and bulk reactions. After a dimensional analysis and an overview of some limiting regimes, we present a set of general-purpose finite-volume solvers, based on OpenFOAM[®], capable of describing an arbitrary number of electrochemical species over multiple interacting (solid or fluid) domains (Icardi and Barnett in F Municchi spnpFoam, 2021. <https://doi.org/10.5281/zenodo.4973896>). We provide a verification of the computational approach for several cases involving electrokinetic flows, reactions between species, and complex geometries. We first present three one-dimensional verification test-cases, and then show the capability of the solver to tackle two- and three-dimensional electrically driven flows and ionic transport in random porous structures. The purpose of this work is to lay the foundation of a general-purpose open-source flexible modelling tool for problems in electrochemistry and electrokinetics at different scales.

Keywords Electrochemical modelling · Stokes-Poisson-Nernst-Planck · OpenFOAM · Porous media · Fluid-solid systems

1 Introduction

Electrokinetic flows is a highly active topic of discussion branching over a multitude of scientific fields. Examples include chloride transport in reinforced concrete [13, 38], ion regulation in biological cells [20, 40], fuel cells [42] and electrochemical energy storage [23, 33], such as batteries and super-capacitors [17, 18, 32]. Consequently, the demand for the numerical modelling of electrokinetic flows, often due to complex geometries or multi-physics barring analytical solutions, spans a great deal of fields. While each real-world example given often comes with its own bespoke problems to consider, they are all describable as specific cases under a generalised model for electrokinetic flow problems. Therefore, the motivation for this paper is to lay the foundational work of a general-purpose open-source computational fluid dynamics (CFD) toolbox for the modelling of electrokinetic

flows [11]. Built in a modular and sequential fashion so additional physics, like steric effects [14, 21, 37] or chemical activity [34], can be subsequently 'grafted' onto the workflow and solved too. As such, we chose the finite-volume CFD package OpenFOAM[®] as our underlying CFD package for implementation.

With electrokinetic flows the involvement of electrical forces leads to a number of interesting phenomena. One such phenomenon is electro-osmosis [1], where an applied electric field induces fluid movement due to formation of electric double layers, see Fig. 2. Another interesting set of phenomena are known as induced-charge electrokinetics. Whilst similar to traditional electrokinetic phenomena, the difference comes from the double layer being induced by an applied electric field [36]. The main difficulty of modelling electrokinetic flows stems from the microscopic scale of phenomena. For applications with large scale domains, solving at the microscopic scale can be computationally taxing. For systems with complex geometries, such as porous media, this difficulty is furthered.

To describe electrokinetic flows we require modelling the ion concentrations, electric potential and fluid veloc-

✉ Matteo Icardi
matteo.icardi@nottingham.ac.uk

¹ University of Nottingham, Nottingham, UK

² Colorado School of Mines, Golden, CO, USA

ity, with the Stokes-Nernst-Planck-Poisson (SPNP) model. In this work, for the velocity field we will, in fact, consider Stokes' flow as, at the micro- and meso-scale, the viscous forces far outweigh the inertial due to small velocities and length scale [35]. The Stokes equation is coupled with the Poisson's equation for the electric potential, relating the electric fields composition to the variation in ion charge density. Finally these are both coupled with the Nernst-Planck equation to describe the ion transport. In using Nernst-Planck, we neglect any ion-ion interaction by assuming the ionic solution is sufficiently dilute.

The Nernst-Planck ionic flux was first formulated in its steady form for a one-dimensional cylinder by Nernst [26]. It was later extended by Planck [30] to a transient setting, furthermore introducing the continuity and Poisson's equations [22]. In doing this the two paved the way in helping develop SPNP, providing a simple yet accurate description of electrokinetic flow for dilute solutions.

A common similarity in many applications containing electrokinetic flows is the involvement of multiple regions, such as fluid electrolyte and solid electrodes in batteries. These regions exchange ions with each other and in some instances contain chemical reactions exchanging mass between species. For batteries these reactions are a necessary process for operational use. However in examples such as chloride corrosion in reinforced concrete, it is a detriment [25]. As such, modelling these reactions has just as much importance as the flow they reside in.

Whilst used extensively in a wide range of physical settings by the research community, SPNP does come with its own caveats. For one, SPNP neglects any ion-ion interactions that may occur by assuming a dilute solution. This may not hold true for solutions with many ionic species [32]. Also, as SPNP is a continuum model, any steric effects are ignored. As such, many efforts have been made extending SPNP to include other physical processes. To cover steric effects, free energy functionals using density functional theory (DFT) accounting for long-range Coulomb correlation and hard sphere (HS) interactions of ions [14, 21, 37] are formulated. Extensions to make Nernst-Planck more thermodynamically consistent under non-equilibrium thermodynamics [8, 18] have also been proposed. To model non-ideal solutions, [34] proposes an added term to the Nernst-Planck ionic flux considering varying chemical activities solved by the Debye-Hückel model.

Whilst these extensions do further the physical realism of the original SPNP model, this often exacerbates other challenges of SPNP. One of the foremost being the non-linear coupling between fields. For example, in [34] the Debye-Hückel model equates the chemical activity of a species to the solutions ionic strength. This in part creates explicit coupling between all ionic concentrations unlike in classic SPNP,

resulting in more complex, often unviable, computational approaches.

The development of numerical solvers for such equations within general PDE and CFD toolboxes is something that has been discussed for decades.

Two common approaches can be taken to solve systems of coupled discretised equations. The first being the so-called block-coupled, with all equations solved at once in a large matrix. Whilst taking a large amount of memory, it upholds the coupling between fields and is numerically robust. The second is the segregated approach and consists in solving each equation separately and in sequence. Since this leads to a decoupling of the equations, appropriate iterative methods [27, 28] must be used to ensure coupling between fields. The advantages of a segregated approach are the lower memory requirements, easy preconditioning of the equations, and their multi-stage structure that allows a better control on the solution procedure. However, block-coupled approaches tend to scale better with the number of processors. When constructing our solvers to model electrokinetic flows, we chose a segregated iterative approach to couple the equations and the different domains.

This work presents a multi-region multi-species SPNP model and discusses its implementation in finite-volumes segregated solvers, built with the OpenFOAM[®] library and released open-source [11]. We present the mathematical model, including a dimensional analysis, and consider multiple solid and fluid regions, with general reaction and interface models.

Whilst other finite-volumes and finite-elements solvers have been developed [3, 19, 29, 39, 41], restrictions such as being designed for specific applications, single domains, dimensionality, absence of reactions, steady state or ignoring the fluid velocity are often made.

Another point of novelty also stems from a general non-linear reaction model, so that various reaction rate models such as Butler-Volmer [33] or the *rate law* [6] can be applied.

This work is organised in the following way. Within §2 we present the governing equations of Stokes-Poisson-Nernst-Planck and what fluid properties are assumed to give accurate flow description. In §3 we perform dimensional analysis to understand the transport regimes possible and how this results in the often used electro-neutrality approximation. For §4 we outline what is required to capture reactions at a multi-region interface given different restriction on ion movement. In §5 the implementation of our solvers for single and multi-region is discussed, as well as the iterative algorithm performed when introducing our reactive conditions. To verify accuracy of our solvers and reactive conditions, we provide necessary numerical examples in §6, with concluding remarks given in §7.

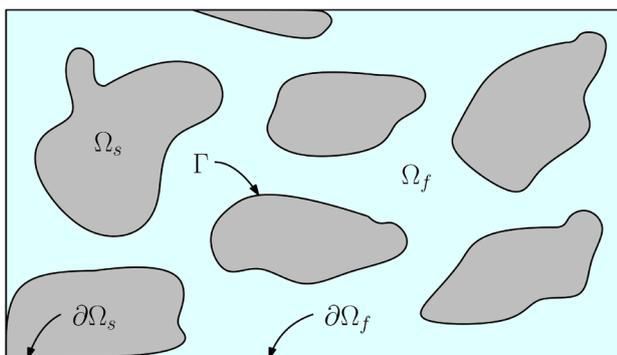


Fig. 1 Graphical representation of the whole domain $\Omega = \Omega_s \cup \Omega_f$ considering two sub-domains: Ω_s a solid region (not necessarily connected), with external boundary $\partial\Omega_s$; and, Ω_f a fluid region with $\partial\Omega_f$ external boundary such that $\partial\Omega = \partial\Omega_s \cup \partial\Omega_f$. Solid–fluid interface denoted as Γ

2 Stokes-Poisson-Nernst-Planck model

Here we discuss the equations that make up Stokes-Poisson-Nernst-Planck, modelling the advective, diffusive and electrostatic forces of an ionic solution. As many real-world applications of electrochemistry involve interacting solids and fluids we consider a multi-domain scenario of a whole domain Ω split, without loss of generality, into two sub-domains Ω_f , a fluid, and Ω_s , a solid, such that $\Omega = \Omega_f \cup \Omega_s$, and with Γ being the solid–fluid interface and $\partial\Omega = \partial\Omega_f \cup \partial\Omega_s$ the external boundaries. See Fig. 1 for a theoretical sketch of the domain Ω . More in general, in some applications and in our computational framework, we have allowed for an arbitrary number of solid and fluid regions, separated by different interfaces. We consider N ionic species, with concentrations and valencies c_i and z_i respectively, and $i = 1, \dots, N$. To describe the ionic transport we must define equations for the electric field \mathbf{E} , ion concentrations c_i within Ω and fluid velocity \mathbf{u} within Ω_f .

2.1 Stokes’ flow

Consider Ω_f , with velocity profile $\mathbf{u}(\mathbf{x}, t)$ governing the advective dynamics of the ions. Assume a negligible Reynolds numbers Re defined by the fluid density ρ_f , characteristic velocity U , characteristic length scale L and dynamic viscosity μ :

$$Re := \frac{\rho_f L U}{\mu} \ll 1,$$

such that viscous forces within Ω_f outweigh the inertial. This common assumption in ionic transport [7, 15, 35] leads to linear Stokes flow. Furthermore we assume the fluid to be

incompressible, i.e.,

$$\mu \nabla^2 \mathbf{u} = \nabla p - \rho_{el} \mathbf{E}, \quad \mathbf{x} \in \Omega_f, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0, \quad \mathbf{x} \in \Omega_f, \tag{2}$$

where $p = p(\mathbf{x}, t)$, $\rho_{el} = \rho_{el}(\mathbf{x}, t) = F \sum_{i=1}^N z_i c_i$ and \mathbf{E} are the static fluid pressure, electric charge density and electric field, respectively, and F is Faraday’s constant. Compared to the standard Stokes equation, we have the presence of the body force term $\rho_{el} \mathbf{E}$, describing the Coulomb forces acted on the fluid by the ions [15, 31]. We neglect any magnetic contribution by assuming our ions move slowly such that \mathbf{E} is irrotational, i.e. $\nabla \times \mathbf{E} = \mathbf{0}$. This body force term may be set to zero if the fluid is unaffected by \mathbf{E} or the fluid is approximated as electrically-neutral, $\rho_{el} = 0$. This will be further discussed in §3.

The first coupling term, between the variables \mathbf{u} , \mathbf{E} and c_i appears here, showing one of the significant difficulties of describing electrokinetic flows. The coupling terms (particularly if non-linear) often add significant numerical difficulties. First of all, they make the velocity field time-dependent. Although by neglecting the time derivative we assume instantaneous relaxation to an equilibrium, and thereby a steady solution, by involving $c_i(\mathbf{x}, t)$ the relaxation becomes tied to the time scale of Nernst-Planck, which is order magnitudes different. For segregated approaches, the disparity of relaxation time scales between Stokes, Poisson and Nernst-Planck—leading to mixed parabolic-elliptic systems—can pose severe instability problems or slow convergence of the coupled system.

2.2 Poisson equation

To model the electric field \mathbf{E} , neglecting magnetic forces, We may then write $\mathbf{E} = -\nabla\phi$ and focus on the electric potential $\phi = \phi(\mathbf{x}, t)$. Assume for each sub-domain their respective electric permittivity ε is spatially constant. From Maxwell’s equations we obtain Poisson’s electrostatic equation denoting variations in ϕ by changes in the charge density ρ_{el} ,

$$\nabla^2 \phi = -\frac{\rho_{el}}{\varepsilon} = -\frac{F}{\varepsilon} \sum_{i=1}^N z_i c_i, \quad \mathbf{x} \in \Omega_{\{f,s\}}, \tag{3}$$

where $\varepsilon = \varepsilon_s$ in the solid and $\varepsilon = \varepsilon_f$ in the fluid. Again we have direct coupling between our variables, here between c_i and ϕ , although this time the former appears linearly in the source term. Like for Stokes flow, this equation depends on time only through the source/coupling terms, in particular the time-dependent ionic concentration $c_i(\mathbf{x}, t)$.

2.3 Ionic transport

Assume the ionic fluid in Ω_f is sufficiently dilute to ignore ion-ion interactions and diffusion is isotropic. Under these assumptions we may use the Nernst-Planck flux [10, 15, 22, 30] as

$$\mathbf{j}_i = \mathbf{j}(c_i, \phi) = \begin{cases} -D_{i,s} (\nabla c_i) & \mathbf{x} \in \Omega_s, \\ -D_{i,f} \left(\nabla c_i + \frac{Fz_i}{RT} c_i \nabla \phi \right) + \mathbf{u} c_i & \mathbf{x} \in \Omega_f, \end{cases} \quad (4)$$

where denote R , T , $D_{i,f}$ and $D_{i,s}$ are, respectively, the ideal gas constant, absolute temperature and diffusion coefficient of species i in the fluid and in the solid. Taking Eq. (4) in conjunction with the continuity equation for mass conservation we arrive at the set of equations modelling transport of c_i ,

$$\frac{\partial c_i}{\partial t} + \nabla \cdot \mathbf{j}_i = 0, \quad \mathbf{x} \in \Omega_{s,f}. \quad (5)$$

As mentioned, we assume a dilute solution to ignore ion-ion interactions. This may not hold true in some cases. Alternatively the Stefan-Maxwell equations [15, 16, 32] may be used in lieu of Nernst-Planck. In short, Stefan-Maxwell balances the driving forces exerted on a species with the frictional forces between species. This introduces cross diffusivities \mathcal{D}_{ij} describing the drag between species i and j due to these frictional forces [15]. The added complexity from such explicit coupling between species is difficult to describe, as the exact description of \mathcal{D}_{ij} is hard to determine [32]. Even in situations where the ionic fluid is bordering on dilute, it is common for Nernst-Planck to still be used [33] due to its simplicity.

To close the system of equations listed here, we must include conditions along the boundaries of Ω_f and Ω_s . We briefly discuss some common choices below and their physical representation. We also require conditions at the interface Γ to describe interaction between sub-domains. This is discussed further below for c_i and ϕ when considering reactions and (non-)conductive interfaces.

2.4 Boundary conditions

Below we list some relevant conditions for different physical situations often seen in electrokinetic problems:

- *Solid walls*: Either a conductive or insulating wall. The no-penetration ($\mathbf{u} \cdot \mathbf{n} = 0$, and zero normal stresses), where \mathbf{n} is the outward unit normal to the boundary, or no-slip condition ($\mathbf{u} = \mathbf{0}$) may be used. Ions may not pass through the wall so their fluxes are zero ($\mathbf{j}_i \cdot \mathbf{n} = 0$). If

conductive we impose a fixed normal current density I_w along the wall ($\mathbf{i} \cdot \mathbf{n} = F \sum_i z_i \mathbf{j}_i \cdot \mathbf{n} = I_w$) or fixed potential ϕ_w . If insulating (non-conductive) we may simply set $I_w = 0$.

- *Permeable membrane and reactive boundaries*: For a membrane we may fix the flux of each species ($\mathbf{j}_i \cdot \mathbf{n} = j_m$), non-zero for those capable of passing through, zero for who cannot. We will discuss the case of reactive boundaries later in §4.2.
- *Inlet/Outlet*: Often used to model an in or outflow of fluid or ions. For a momentum driven inlet, pressure or flow velocity may be fixed ($\mathbf{u} = \mathbf{u}_{in}$, $p = p_{in/out}$). For ions we may apply fixed fluxes ($\mathbf{j}_i \cdot \mathbf{n} = j_{in/out}$), resulting in a fixed electric current ($\mathbf{i} \cdot \mathbf{n} = I_{in/out}$). More complicated is to determine suitable boundary conditions for the potential. Typically either a Dirichlet $\phi = \phi_{in/out}$ or a Neumann condition can be imposed. The latter imposes the total electric current to be either equal to zero or a fixed value.
- *Periodicity*: when dealing with large (quasi-)periodic (or homogeneous) structures, it is often impossible to solve for the entire domain of interest. In these cases, smaller representative unit cells can be solved with (quasi-)periodic external boundary conditions [5] are imposed. In these cases, an additional driving forces (as a bulk source term or a modification of the periodic BC) need to be added.

3 Dimensional analysis

When dealing with systems of coupled transport equations it is useful to perform a dimensional analysis to better understand the relationships between the different transport phenomena and identify the limiting regimes and possible approximations. We denote dimensionless variables with a hat symbol, e.g \hat{x} , and reference values with a bar unless stated otherwise. We use the reference values L , U , $\bar{\phi}$ and \bar{c}_i for the length scale, velocity, electric potential and concentrations respectively. For pressure we take $\bar{p} = \frac{\mu U}{L}$ as this the appropriate form when under Stokes flow. For time, we take the diffusive timescale $\bar{t} = \frac{L^2}{D_i}$. With these choices and

defining $\bar{C} = \frac{1}{2} \sum_{i=1}^N z_i^2 \bar{c}_i$ as the reference total ionic strength, we obtain the dimensionless variables:

$$\hat{x} = \frac{x}{L}, \quad \hat{t} = \frac{D_i t}{L^2}, \quad \hat{\mathbf{u}} = \frac{\mathbf{u}}{U}, \quad \hat{\phi} = \frac{\phi}{\bar{\phi}},$$

$$\hat{c}_i = \frac{c_i}{\bar{c}_i}, \quad \hat{p} = \frac{pL}{\mu U}. \quad (6)$$

Substituting these dimensionless variables into Eqs. (1), (3) and (5) the dimensionless system of equations are

$$\frac{D_i \bar{c}_i}{L^2} \frac{\partial \hat{c}_i}{\partial \hat{t}} + \hat{\nabla} \cdot \left(-D_i \left(\frac{\bar{c}_i}{L^2} \hat{\nabla} \hat{c}_i + \frac{F \bar{\phi} \bar{c}_i}{L^2 RT} \hat{c}_i z_i \hat{\nabla} \hat{\phi} \right) + \frac{U \bar{c}_i}{L} \hat{c}_i \hat{\mathbf{u}} \right) = 0, \tag{7}$$

$$\frac{\bar{\phi}}{L^2} \hat{\nabla}^2 \hat{\phi} = -\frac{F}{\varepsilon} \sum_{i=1}^N z_i \hat{c}_i \bar{c}_i, \tag{8}$$

$$\frac{\mu U}{L^2} \hat{\nabla}^2 \hat{\mathbf{u}} = \frac{\mu U}{L^2} \hat{\nabla} \hat{p} + \frac{F \bar{\phi}}{L} \hat{\nabla} \hat{\phi} \sum_{i=1}^N z_i \hat{c}_i \bar{c}_i. \tag{9}$$

The underlying dimensionless numbers may be found by dividing all other terms by the reference values of one term. For Eqs. (7) to (9) we divide by $\frac{D_i \bar{c}_i}{L^2}$, $\frac{2\bar{C}F}{\varepsilon}$ and $\frac{2\bar{C}\mu U}{2\bar{C}L^2}$ respectively, where recall \bar{C} is the reference ionic strength needed to resolve the issue of not being able to factor out the reference concentrations \bar{c}_i . As such, the dimensionless equations become:

$$\frac{\partial \hat{c}_i}{\partial \hat{t}} + \hat{\nabla} \cdot \left(-\left(\hat{\nabla} \hat{c}_i + \mathcal{N} \hat{c}_i z_i \hat{\nabla} \hat{\phi} \right) + \text{Pe}_i \hat{\mathbf{u}} \hat{c}_i \right) = 0, \tag{10}$$

$$\mathcal{L}^2 \hat{\nabla}^2 \hat{\phi} = -\frac{\sum_{i=1}^N \hat{c}_i \bar{c}_i z_i}{2\bar{C}}, \tag{11}$$

$$\hat{\nabla}^2 \hat{\mathbf{u}} = \hat{\nabla} \hat{p} + \frac{\mathcal{P}}{2\bar{C}} \sum_{i=1}^N \hat{c}_i \bar{c}_i z_i \hat{\nabla} \hat{\phi}, \tag{12}$$

where we have defined four dimensionless numbers \mathcal{N} , Pe , \mathcal{L} and \mathcal{P} . Note that whilst Eq. (7) had four terms, we only arrive at two numbers due to our choice of reference time. The same can be said with Eq. (12) and chosen reference pressure. The result, four numbers for the three equations with forms given in Eq. (13) as:

$$\mathcal{N} = \frac{F \bar{\phi}}{RT} = \frac{e \bar{\phi}}{k_B T}, \quad \text{Pe} = \frac{UL}{D_i}, \tag{13}$$

$$\mathcal{L}^2 = \frac{\varepsilon k_B T}{2N_A e^2 L^2 \bar{C}}, \quad \mathcal{P} = \frac{2FL\bar{\phi}\bar{C}}{\mu U},$$

written in terms of the aforementioned reference values, where we denote e , k_B and N_A to be the elementary charge, Boltzmann’s constant and Avogadro’s number respectively. \mathcal{N} represents the ratio of electrostatic over diffusive forces, with $\mathcal{N} \ll 1$ indicating diffusion is dominant. We also obtain the Péclet number Pe , i.e., the ratio of advective over diffusive phenomena, and $\mathcal{L} = \frac{\lambda_D}{L}$ the dimensionless Debye length, with the dimensional form λ_D defined as

$$\lambda_D = \sqrt{\frac{\varepsilon k_B T}{2N_A e^2 \bar{C}}}, \tag{14}$$

approximating the distance at which a charge’s electrostatic effect persists. Typically $\mathcal{L} \ll 1$, stating how the Debye length is much smaller than the reference length L . Finally

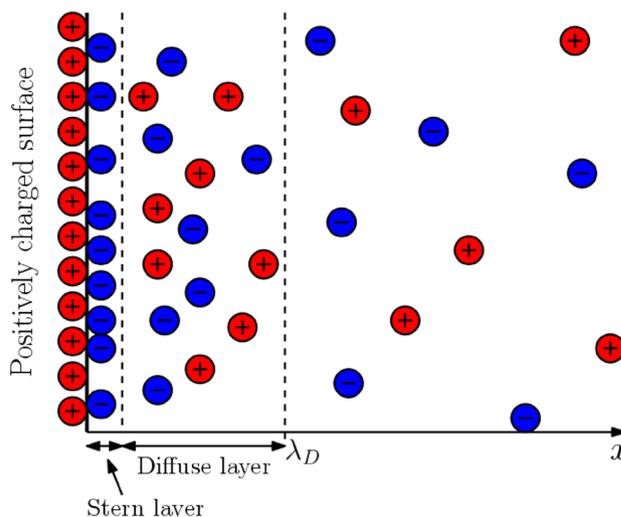


Fig. 2 A graphical representation of the electric double layer along a positively charged surface, comprised of a thin layer of adsorbed negative ions (Stern layer) and loosely connected positive and negative ions (Diffuse layer)

we have \mathcal{P} denoting the ratio of viscous and electric forces upon our fluid. λ_D is also the approximate width of a common electrokinetic phenomena known as the electric double layer (EDL), see Fig. 2, that forms on boundaries. The EDL consists first of ions adsorbed at the boundary, known as the Stern layer, and another of free ions moved by electrical attraction and diffusive motion by the Stern layer, deemed the diffuse layer.

For sufficiently thin EDLs there is a common model reduction known as the electro-neutrality assumption, often employed in development of macroscopic models [17, 33, 35]. This reduction can in fact be determined through dimensional analysis and will be briefly discussed next.

3.1 Asymptotics and electro-neutrality

The electro-neutrality approximation states that for a sufficiently dilute electrolyte all charges of ionic species within the solution roughly cancel each other out, leaving the solution electrically neutral. Electro-neutrality for a solution containing N species is defined as

$$\sum_{i=1}^N z_i c_i = 0, \tag{15}$$

and is often used as model reduction when modelling ionic flows. Where Eq. (15) becomes invalid however is in the thin charged double layers, or EDL, mentioned above and often seen in real-world settings.

To get an understanding of where electro-neutrality comes from and how it relates to EDL formation, we perform an asymptotic analysis. Without any lack of generality, we limit

here to a one-dimensional electrolyte and a binary electrolyte. In the numerical framework described above an arbitrary number of ionic species in three-dimensions can be considered. Let $x \in \Xi$ where $\Xi = [0, 1]$ is a binary electrolyte solution with ionic concentrations c_1 and c_2 of opposing valencies. For now we omit any boundary conditions, only requiring those chosen result in a boundary layer formation near $x = 0$. To arrive at Eq. (15) we start by considering the asymptotics of the outer (bulk) layer away from $x = 0$. For simplicity we only consider the leading order terms.

Outer (bulk) layer To arrive at an asymptotic leading order solution in the bulk layer of Ξ we take the following expansions of \hat{u} , $\hat{\phi}$, \hat{p} , \hat{c}_1 and \hat{c}_2 . These denote the fluid velocity, electric potential, pressure and ion concentrations respectively, all dimensionless. Expansions are taken in powers of $\epsilon = \mathcal{L}^2$, the squared dimensionless Debye screening length since $\epsilon \ll 1$, or, $\lambda_D \ll L$:

$$\hat{\phi} = \phi^{(0)} + \epsilon\phi^{(1)} + \mathcal{O}(\epsilon^2), \tag{16}$$

$$\hat{c}_i = c_i^{(0)} + \epsilon c_i^{(1)} + \mathcal{O}(\epsilon^2), \quad i = 1, 2, \tag{17}$$

$$\hat{u} = u^{(0)} + \epsilon u^{(1)} + \mathcal{O}(\epsilon^2), \tag{18}$$

$$\hat{p} = p^{(0)} + \epsilon p^{(1)} + \mathcal{O}(\epsilon^2). \tag{19}$$

Substituting these expansions into the transport Eqs. (10)–(12) for one-dimension, alongside the incompressibility condition, we arrive at

$$\begin{aligned} \mathcal{O}(\epsilon^2) = & \frac{\partial}{\partial t} (c_i^{(0)} + \epsilon c_i^{(1)}) + \frac{\partial}{\partial x} \left(\text{Pe}_i [u^{(0)} + \epsilon u^{(1)}] \right. \\ & \left. [c_i^{(0)} + \epsilon c_i^{(1)}] - \frac{\partial}{\partial x} (c_i^{(0)} + \epsilon c_i^{(1)}) \right. \\ & \left. - \mathcal{N} z_i (c_i^{(0)} + \epsilon c_i^{(1)}) \frac{\partial}{\partial x} (\phi^{(0)} + \epsilon \phi^{(1)}) \right), \quad i = 1, 2, \end{aligned} \tag{20}$$

$$\mathcal{O}(\epsilon^2) = \epsilon \frac{\partial^2}{\partial x^2} (\phi^{(0)} + \epsilon \phi^{(1)}) + \frac{1}{2\bar{C}} \sum_{i=1}^2 z_i \bar{c}_i (c_i^{(0)} + \epsilon c_i^{(1)}), \tag{21}$$

$$\begin{aligned} \mathcal{O}(\epsilon^2) = & -\frac{\partial^2}{\partial x^2} (u^{(0)} + \epsilon u^{(1)}) + \frac{\partial}{\partial x} (p^{(0)} + \epsilon p^{(1)}) \\ & + \frac{\mathcal{P}}{2\bar{C}} \frac{\partial}{\partial x} (\phi^{(0)} + \epsilon \phi^{(1)}) \sum_{i=1}^2 z_i \bar{c}_i (c_i^{(0)} + \epsilon c_i^{(1)}), \end{aligned} \tag{22}$$

$$\mathcal{O}(\epsilon^2) = \frac{\partial}{\partial x} (u^{(0)} + \epsilon u^{(1)}). \tag{23}$$

By considering only the leading order terms i.e., terms of $\mathcal{O}(1)$, these equations reduce to,

$$\begin{aligned} \frac{\partial}{\partial t} c_i^{(0)} + \frac{\partial}{\partial x} \left(\text{Pe}_i u^{(0)} c_i^{(0)} - \frac{\partial}{\partial x} c_i^{(0)} - \mathcal{N} z_i c_i^{(0)} \frac{\partial}{\partial x} \phi^{(0)} \right) \\ = 0, \quad i = 1, 2, \end{aligned} \tag{24}$$

$$0 = z_1 \bar{c}_1 c_1^{(0)} + z_2 \bar{c}_2 c_2^{(0)}, \tag{25}$$

$$\frac{\partial^2}{\partial x^2} u^{(0)} = \frac{\partial}{\partial x} p^{(0)}, \tag{26}$$

$$\frac{\partial}{\partial x} u^{(0)} = 0. \tag{27}$$

Note how Eq. (25) is the electro-neutrality approximation mentioned before, for a binary solution, and a direct consequence of the asymptotics. This implies said electro-neutrality is only accurate up to leading order. What’s more the electric body force of Stokes vanishes as a consequence of Eq. (25). To close the system an equation for the leading order potential $\phi^{(0)}$ is required. This is possible by multiplying Eq. (24) by their respective valencies z_i and reference values \bar{c}_i , summing over i , and utilizing Eq. (25). Resulting in the equation,

$$\begin{aligned} \frac{\partial}{\partial x} \left(u^{(0)} \sum_i \text{Pe}_i z_i \bar{c}_i c_i^{(0)} - \sum_i z_i \bar{c}_i \frac{\partial}{\partial x} c_i^{(0)} \right. \\ \left. - \mathcal{N} \sum_i z_i^2 \bar{c}_i c_i^{(0)} \frac{\partial}{\partial x} \phi^{(0)} \right) = 0, \end{aligned} \tag{28}$$

which represents a steady equation for $\phi^{(0)}$. This can be interpreted as a modified Ohm’s law with new conductivity accounting for all contributions to the total current: advective, diffusive and electrical, from left to right.

Now that we have constructed the asymptotic solution, up to $\mathcal{O}(1)$, for the outer (bulk) layer of Ξ , we move on to determine the inner (boundary) layer asymptotics. As mentioned at the start, we state said inner layer formation is near $x = 0$ of Ξ . Much like the outer layer, we make no case of boundary conditions, only that a boundary layer near $x = 0$ forms as a result of them.

Inner solution:

To construct a solution for the inner layer near $x = 0$ of Ξ we define the variable y to span the inner layer and be a ‘fast’ variable counterpart to x , changing more rapidly,

$$y = \frac{x}{\sqrt{\epsilon}}, \quad y \in [0, \infty). \tag{29}$$

By the definition of $x \in \Xi$, y therefore has domain $[0, \infty)$, as $y = 0$ for $x = 0$ and $y \rightarrow \infty$ for $x \rightarrow 1$, since $\epsilon \ll 1$. Such change of variable gives derivatives via chain rule as:

$$\frac{\partial}{\partial x} = \frac{1}{\sqrt{\epsilon}} \frac{\partial}{\partial y}, \quad \frac{\partial^2}{\partial x^2} = \frac{1}{\epsilon} \frac{\partial^2}{\partial y^2}. \tag{30}$$

Substitution of the above derivatives into the original transport equations Eqs. (10)–(12) results in,

$$\frac{\partial \hat{c}_i}{\partial \hat{t}} + \frac{1}{\sqrt{\epsilon}} \frac{\partial}{\partial y} \left(\text{Pe}_i \hat{u} \hat{c}_i - \frac{1}{\sqrt{\epsilon}} \frac{\partial \hat{c}_i}{\partial y} - \frac{1}{\sqrt{\epsilon}} \mathcal{N} z_i \hat{c}_i \frac{\partial \hat{\phi}}{\partial y} \right) = 0, \tag{31}$$

$$\frac{\partial^2 \hat{\phi}}{\partial y^2} = -\frac{1}{2\bar{C}} \sum_{i=1}^2 z_i \bar{c}_i \hat{c}_i, \tag{32}$$

$$\frac{1}{\epsilon} \frac{\partial^2 \hat{u}}{\partial y^2} = \frac{1}{\sqrt{\epsilon}} \frac{\partial \hat{p}}{\partial y} + \frac{1}{\sqrt{\epsilon}} \frac{\mathcal{P}}{2\bar{C}} \frac{\partial \hat{\phi}}{\partial y} \sum_{i=1}^2 z_i \bar{c}_i \hat{c}_i, \tag{33}$$

$$\frac{\partial \hat{u}}{\partial y} = 0, \tag{34}$$

where all variables \hat{u} , \hat{p} , $\hat{\phi}$ and \hat{c}_i are in terms of y e.g., $\hat{c}_i = \hat{c}_i(y, t)$. Just like the outer layer, we substitute the asymptotic expansions Eqs. (16)–(19) into the above equations to obtain,

$$\begin{aligned} \frac{\partial}{\partial \hat{t}} \left(c_i^{(0)} + \epsilon c_i^{(1)} \right) + \frac{1}{\sqrt{\epsilon}} \frac{\partial}{\partial y} \left(\text{Pe}_i \left[u^{(0)} + \epsilon u^{(1)} \right] \left[c_i^{(0)} + \epsilon c_i^{(1)} \right] \right. \\ \left. - \frac{1}{\sqrt{\epsilon}} \frac{\partial}{\partial y} \left(c_i^{(0)} + \epsilon c_i^{(1)} \right) \right. \\ \left. - \mathcal{N} z_i \frac{1}{\sqrt{\epsilon}} \left(c_i^{(0)} + \epsilon c_i^{(1)} \right) \frac{\partial}{\partial y} \left(\phi^{(0)} + \epsilon \phi^{(1)} \right) \right) = \mathcal{O}(\epsilon^2), \end{aligned} \tag{35}$$

$i = 1, 2,$

$$\frac{\partial^2}{\partial y^2} \left(\phi^{(0)} + \epsilon \phi^{(1)} \right) = -\frac{1}{2\bar{C}} \sum_{i=1}^2 z_i \bar{c}_i \left(c_i^{(0)} + \epsilon c_i^{(1)} \right) + \mathcal{O}(\epsilon^2), \tag{36}$$

$$\begin{aligned} \frac{1}{\epsilon} \frac{\partial^2}{\partial y^2} \left(u^{(0)} + \epsilon u^{(1)} \right) = \frac{1}{\sqrt{\epsilon}} \frac{\partial}{\partial y} \left(p^{(0)} + \epsilon p^{(1)} \right) \\ + \frac{1}{\sqrt{\epsilon}} \frac{\mathcal{P}}{2\bar{C}} \frac{\partial}{\partial y} \left(\phi^{(0)} + \epsilon \phi^{(1)} \right) \sum_{i=1}^2 z_i \bar{c}_i \left(c_i^{(0)} + \epsilon c_i^{(1)} \right) \\ + \mathcal{O}(\epsilon^2), \end{aligned} \tag{37}$$

$$\frac{\partial}{\partial y} \left(u^{(0)} + \epsilon u^{(1)} \right) = \mathcal{O}(\epsilon^2). \tag{38}$$

Considering only the leading order terms of the four equations above we arrive at the following leading order set of equations for the inner layer:

$$\frac{\partial}{\partial y} \left(-\frac{\partial}{\partial y} c_i^{(0)} - \mathcal{N} z_i c_i^{(0)} \frac{\partial}{\partial y} \phi^{(0)} \right) = 0, \quad i = 1, 2, \tag{39}$$

$$\frac{\partial^2}{\partial y^2} \phi^{(0)} = -\frac{1}{2\bar{C}} \sum_{i=1}^2 z_i \bar{c}_i c_i^{(0)}, \tag{40}$$

$$\frac{\partial^2}{\partial y^2} u^{(0)} = 0, \tag{41}$$

$$\frac{\partial}{\partial y} u^{(0)} = 0. \tag{42}$$

Like the outer layer equations, to close the system we need another equation, this time for $p^{(0)}$. Considering the $\mathcal{O}(\epsilon^{(1/2)})$ terms of Eq. (37) we can retrieve such an equation as

$$\frac{\partial}{\partial y} p^{(0)} + \frac{\mathcal{P}}{2\bar{C}} \frac{\partial}{\partial y} \phi^{(0)} \sum_{i=1}^2 z_i \bar{c}_i c_i^{(0)} = 0. \tag{43}$$

As mentioned earlier, when considering situations involving multiple sub-domains we must also have appropriate interface conditions. In many real applications of electrokinetic flows there are chemical reactions at interfaces between sub-domains, exchanging mass across the ionic species involved. In these cases, neglecting the EDL might lead to significant errors. In the next section we will consider such reactions occurring on our interface Γ , formulating appropriate conditions to capture them whilst retaining mass conservation.

4 Multi-domain formulation and reactions

Here we formulate conditions to model heterogeneous reactions which are crucial for many electrochemical and electrokinetic problems. We write a general reaction rate that ensures total mass conservation and apply it to form reactive interface conditions, where we consider the scenarios of species that exist in the whole domain (*unrestricted*) or only in a specific sub-domain (*restricted*). We then discuss conditions on ϕ when the interface is conductive or non-conductive.

4.1 Reaction model

Consider a general elementary reaction transferring mass between reactants $i = J, \dots, K$ and products $i = K + 1, \dots, M$ with exchanged molar masses B_i and valencies z_i . We denote $v_i > 0$ to be the stoichiometric coefficients determining the number of moles of species i is lost or gained and n the number of released ($n < 0$) or absorbed ($n > 0$) electrons, alongside the electron mass e^- . The mass balance of the reaction reads:

$$\sum_{i=J}^K v_i B_i + n e^- = \sum_{i=K+1}^M v_i B_i, \tag{44}$$

$$n = \sum_{i=K+1}^M z_i v_i - \sum_{i=J}^K v_i z_i. \tag{45}$$

To formulate Eq. (44) as conditions on Γ we first determine the rate r_i at which each species in our reaction is exchanging mass. More complex reactions involving several intermediate reactions can be decomposed into elementary steps, i.e a set

of elementary reactions. Typically the overall reaction rate is then given by the rate of the slowest elementary reaction [6]. One option to determine the rate is the *law of mass action*, also known as the *rate law* [2, 6]. Given an ideal solution and reaction involving chemical species $[j]$ with stoichiometric coefficients v_j , at dynamic equilibrium

$$\sum_{\text{reactants}} v_j[j] \rightleftharpoons \sum_{\text{products}} v_j[j], \quad (46)$$

where the net reaction rate (density) r' with units $[\text{mol}/\text{m}^2 \text{ s}]$, is given by:

$$r' = k_f \prod_j \left(c_j^{v_j} \right)_{\text{react}} - k_r \prod_j \left(c_j^{v_j} \right)_{\text{prod}}. \quad (47)$$

Here k_f, k_r are the rate constants for the forward and reverse reaction and can be empirically modelled by the Arrhenius equation [6, 31]. Note that r' is always positive, so we cannot simply take $r_i = r'$ as we must allow $r_i < 0$ for reactant species. Instead, we define the coefficient α_i as

$$\alpha_i = \begin{cases} -1 & \text{reactants,} \\ +1 & \text{products,} \end{cases} \quad (48)$$

and, multiplying by r' and v_i , we obtain:

$$r_i = \alpha_i v_i r' = \alpha_i v_i \left(k_f \prod_{j=J}^K \left(c_j^{v_j} \right)_{\text{react}} - k_r \prod_{j=K+1}^M \left(c_j^{v_j} \right)_{\text{prod}} \right), \quad (49)$$

$i = J, \dots, M.$

For species not involved in the reaction we set $r_i = 0$. Assuming a closed reactive system, i.e no trace ion species, we can represent the mass conservation as the balance across reaction rates r_i :

$$\sum_{i=1}^N m_i r_i = 0. \quad (50)$$

We weight by the molar masses m_i to convert from moles (a non-conserved quantity between species) to grams. We use the *rate law* as the example here as it is valid for many reactions [6]. It is also the more general form for the commonly used Butler-Volmer equation for faradaic reactions, which uses the energy dependence of k_f and k_r to give explicit electric potential dependency. It is important to notice that the units of measure of the reaction rate are $[\text{mol}/\text{s}]$ for bulk reaction and $[\text{m mol}/\text{s}]$ for surface reactions. Therefore, for the case of linear reactions, the reaction constants k_i can be either $[1/\text{s}]$ or $[\text{m}/\text{s}]$.

4.2 Interface conditions

The reaction models above can be applied in the bulk or on a interface. Here we apply them on Γ for ϕ and c_i given conductive or non-conductive interface and surface reactions respectively. We employ general reaction rates r_i to balance the ionic fluxes \mathbf{j}_i through Γ with mass exchanged by the reaction. For ϕ we use conservation of charge to find conditions on the current passing through Γ when acting conductively or non-conductively.

4.2.1 Interface conditions for ions concentration

Here we outline reactive conditions along Γ to model Eq. (44). We use a set of general, possibly non-linear, reaction rates r_i for $i = J, \dots, M$ of ion species involved in Eq. (44). We make no assumptions on the form of r_i , other than making sure Eq. (50) holds true. For species not involved we simply take flux continuity. the results is jump conditions in ionic flux between Ω_s and Ω_f , equal to their respective reaction rates r_i :

$$\left[\mathbf{j}_i \cdot \mathbf{n} \right]_{\Gamma_f}^{\Gamma_s} = \begin{cases} r_i & \mathbf{x} \in \Gamma \text{ for } i = J, \dots, M, \\ 0 & \mathbf{x} \in \Gamma \text{ for } i \neq J, \dots, M. \end{cases} \quad (51)$$

Here we denote $[\mathbf{j}_i \cdot \mathbf{n}]_{\Gamma_f}^{\Gamma_s}$ to be the normal flux evaluated at Γ from Ω_f 's side. Alongside this we assume continuity of concentrations to provide our second condition:

$$c_i|_{\Gamma_s} = c_i|_{\Gamma_f}, \quad \mathbf{x} \in \Gamma. \quad (52)$$

So, for all species involved in the reaction eq. (44) we have the difference in ionic flux from Ω_s and Ω_f to be the rate r_i of the respective species. In some scenarios one or more of the ion species may be restricted to reside in a single sub-domain of Ω . This can however be easily modelled by simply modifying the conditions of those restricted species, setting both \mathbf{j}_i and c_i to be zero in the inaccessible domains.

4.2.2 Interface conditions for the potential

We consider conditions for the electric potential ϕ given two situations. Suppose our interface Γ is acting as a conductor such that we see a electric current flowing through. Assume we are under the most general case where all ion species are unrestricted. Conservation of charge and Ohm's law then implies the change in current through Γ is proportional to the sum of changes in ionic fluxes:

$$F \sum_{i=1}^N z_i \left[\mathbf{j}_i \cdot \mathbf{n} \right]_{\Gamma_f}^{\Gamma_s} = \left[\sigma \nabla \phi \cdot \mathbf{n} \right]_{\Gamma_f}^{\Gamma_s}, \quad \mathbf{x} \in \Gamma. \quad (53)$$

Here we denote σ to be the electric conductivity which is discontinuous across the interface. Alongside this we assume continuity of the potential:

$$\phi|_{\Gamma_s} = \phi|_{\Gamma_f}, \quad \mathbf{x} \in \Gamma. \tag{54}$$

We see therefore that the conditions on the current are closely linked to the previous reactive conditions. As if no reaction occurs at Γ , we simply have continuity of current due to ionic flux continuity. If a reaction is present, then eq. (51) tells us our jump in current is proportional to the sum of reaction rates r_i alongside their respective charge numbers z_i .

If Γ is non-conductive instead we fix separately the current on each side of Γ and no longer impose continuity:

$$F \sum_{i=1}^N z_i [\mathbf{j}_i \cdot \mathbf{n}]_{\Gamma_f} = [\sigma \nabla \phi \cdot \mathbf{n}]_{\Gamma_f}, \quad \mathbf{x} \in \Gamma, \tag{55}$$

$$F \sum_{i=1}^N z_i [\mathbf{j}_i \cdot \mathbf{n}]_{\Gamma_s} = [\sigma \nabla \phi \cdot \mathbf{n}]_{\Gamma_s}, \quad \mathbf{x} \in \Gamma. \tag{56}$$

5 Numerical implementation

As we have mentioned our goal is to construct simple, effective numerical solvers to the Stoke-Poisson-Nernst-Planck (SPNP) model at the pore scale. To do so, we employ the computational fluid dynamics (CFD) package OpenFOAM® due to its open-source nature, large active community and robust handling of complex geometries. OpenFOAM® is a finite volume library for general unstructured mesh. Coupled equations are solved iteratively in a segregated/splitted approach, solving sequentially each discretised equation. This removes the explicit coupling between equations, as well as the need of linearising multi-linear terms (terms linear in each variable but where multiple variables appear) at the expense of having internal iterations. These would be unavoidable also if we adopted a monolithic approach due to the non-linearities of the model.

Two separate solvers have been implemented. The first, *pnpFoam*, models electrokinetic flow of a single ionic fluid, modelled by SPNP. The second, *pnpMultiFoam*, is more generalised, modelling a general set of ionic fluids and solids following SPNP and diffusion respectively. Furthermore we developed the numerical counterpart, named *mappedChemicalKinetics*, to the reactive conditions in §4.2.1 for the case of a binary reaction. In this section we will describe in detail the structure of the solvers and the corresponding boundary conditions.

5.1 Single- and multi-domain solvers

OpenFOAM® discretise each equation using a sparse matrix \mathbf{M} with entries relating to the cell centres of the mesh. For example, for the fluid momentum the discrete form reads:

$$\mathbf{M}[\mathbf{u}] = -\nabla p - \rho_{el} \nabla \phi, \tag{57}$$

where $[\mathbf{u}]$ is the vector of unknown velocities at cell centres and \mathbf{M} the coefficient matrix scaling the effect of neighbouring cell velocities. Note the right hand terms are left as source terms and are computed explicitly and the gradients operators can be discretised with different schemes (details about the schemes will be presented in the results section). To solve for the pressure–velocity coupling we employ the PIMPLE algorithm with an extra term due to the electric body force. This works by decomposing \mathbf{M} into its diagonal, A , and off-diagonal, \mathbf{H} , parts $\mathbf{M}[\mathbf{u}] = \mathbf{A}\mathbf{u} - \mathbf{H}$. This leads to the velocity correction equation:

$$\mathbf{u} = \frac{\mathbf{H}}{A} - \frac{1}{A} \nabla p - \frac{1}{A} \rho_{el} \nabla \phi. \tag{58}$$

Interpolating \mathbf{u} to the cell faces and taking the dot product with cell face area vectors \mathbf{S}_f leads to the flux U correction equation,

$$U = \mathbf{u}_f \cdot \mathbf{S}_f = \left(\frac{\mathbf{H}}{A} \right)_f \cdot \mathbf{S}_f - \left(\frac{1}{A} \right)_f \mathbf{S}_f \cdot \nabla_f^\perp p^{n+1} - \left(\frac{\rho_{el} \nabla \phi}{A} \right)_f \cdot \mathbf{S}_f, \tag{59}$$

where subscript f denotes values at cell faces. Discretising the incompressibility condition $\nabla \cdot \mathbf{u} = 0$ gives $\nabla \cdot U = 0$ which when applied to the flux correction equation forms the pressure correction equation:

$$\nabla \cdot \left(\left[\frac{\mathbf{H}}{A} \right]_f \nabla p \right) = \nabla \cdot \left(\left[\frac{\mathbf{H}}{A} \right]_f - \left[\frac{\rho_{el} \nabla \phi}{A} \right]_f \right). \tag{60}$$

This is solved iteratively until convergence. In an external loop, momentum and pressure equations are coupled with the concentration and potential equations. When the mesh is highly skewed, or for complex discretisation schemes with implicit-explicit terms, additional iterations can be added for each single equation. The pseudo-code algorithm for our single ionic fluid solver *pnpFoam* is presented in Algorithm 1.

Algorithm 1: *pnpFoam* algorithm

```

Construct fields:  $\mathbf{u}$ ,  $p$ ,  $\phi$ ,  $c_i$ ;
while  $t < \text{end time}$  do
  while !converged do
    Initialise  $\mathbf{u}^{n+1}$ ,  $p^{n+1}$  to previous values:  $\mathbf{u}^{n+1} = \mathbf{u}^n$ ,  $p^{n+1} = p^n$ ;
    Solve momentum equation for  $\mathbf{u}^{n+1}$ :  $-\mu \nabla^2 \mathbf{u}^{n+1} + \rho_{\text{el}} \nabla \phi = -\nabla p^{n+1}$ ;
    while Pressure correction do
      Solve pressure correction equation:  $\nabla \cdot \left[ \left( \frac{1}{A} \right)_f \nabla p^{n+1} \right] = \nabla \cdot \left( \left( \frac{\mathbf{H}}{A} \right)_f - \left( \frac{\rho_{\text{el}} \nabla \phi}{A} \right)_f \right)$ ;
      Correct flux:  $U^{n+1} = \mathbf{u}_f \cdot \mathbf{S}_f = \left( \frac{\mathbf{H}}{A} \right)_f \cdot \mathbf{S}_f - \left( \frac{1}{A} \right)_f \mathbf{S}_f \cdot \nabla_f^\perp p^{n+1} - \left( \frac{\rho_{\text{el}} \nabla \phi}{A} \right)_f \cdot \mathbf{S}_f$ ;
      Correct velocity:  $\mathbf{u}^{n+1} = \frac{\mathbf{H}}{A} - \frac{1}{A} \nabla p^{n+1} - \frac{\rho_{\text{el}} \nabla \phi}{A}$ 
    end
    while Non-orthogonal correction do
      Solve Poisson's:  $-\varepsilon \nabla^2 \phi = \rho_{\text{el}}$ ;
    end
    for All  $c_i$ 's do
      while Non-orthogonal correction do
        Solve Nernst-Planck:  $\frac{\partial c_i}{\partial t} + \nabla \cdot \left( \mathbf{u} c_i - D_i \left( \nabla c_i + \frac{F}{RT} z_i c_i \nabla \phi \right) \right) = 0$ ;
      end
    end
  end
  Increment time:  $t \rightarrow t + \Delta t$ 
end

```

The same method is employed within the algorithm of *pnpMultiFoam*, our solver for ionic transport over a general set of ionic fluids and solids, where each region is solved separately, and an additional outer loop is added to ensure the coupling between regions. The algorithm is presented in Algorithm 2. Both solvers here are presented for the case of non-electroneutral solution. In case the electroneutrality is assumed, the algorithm is slightly modified to solve a modified potential equation Eq. (28) (with ionic conductivity instead of permittivity) and with the last species calculated to ensure electro-neutrality.

5.2 Boundary and interface conditions

To make use of the object orientation of OpenFOAM[®], all conditions are first reformulate as effective Robin conditions. We consider here, as an example, the inhomogeneous Robin BC for a variable c with coefficients D^* , K^* and F^* along a boundary Γ with normal \mathbf{n} :

$$D^* \nabla_{\mathbf{n}} c|_{\Gamma} = -K^* c|_{\Gamma} - F^*. \quad (61)$$

In OpenFOAM[®] the boundary values $c|_{\Gamma}$ and $\nabla_{\mathbf{n}} c|_{\Gamma}$ are approximated using the value of the cell centres with faces along Γ :

$$c|_{\Gamma} \approx \alpha_1 c_c + \alpha_2, \quad (62)$$

$$\nabla_{\mathbf{n}} c|_{\Gamma} \approx \alpha_3 c_c + \alpha_4 \approx B(c_f - c_c). \quad (63)$$

Here c_c and c_f are the values of c at the cell centre and face respectively. The α 's are the interpolation weights, and B is the inverse distance between the cell centre and the boundary. We can then rearrange Eq. (61) using Eqs. (62) and (63) to find the α values that allow us to approximate Eq. (61) using the cell centres c_c :

$$\begin{aligned} \alpha_1 &= \frac{D^* B}{D^* B - K^*}, \alpha_2 = \frac{F^*}{D^* B - K^*}, \alpha_3 \\ &= \frac{K^* B}{D^* B - K^*}, \alpha_4 = \frac{F^* B}{D^* B - K^*}. \end{aligned} \quad (64)$$

This forms the basis of the *Robin* BC implemented. Reformulating all other conditions into a form like Eq. (61) lets us reuse the same equations in Eq. (64) to approximate all other conditions.

The interface conditions are implemented as a derived class named *mappedChemicalKinetics*. If we consider here the limiting case of two reacting species c_s and c_f , restricted to their respective sub-domains Ω_s and Ω_f respectively. Along the interface Γ connecting the two sub-domains is the reac-

Algorithm 2: *pnpMultiFoam* algorithm

```

Construct fields:  $\mathbf{u}$ ,  $p$ ,  $\phi_f$ ,  $\phi_s$ ,  $c_{i,f}$ ,  $c_{i,s}$ ;
while  $t < \text{end time}$  do
  for All fluid regions do
    Initialise  $\mathbf{u}^{n+1}$ ,  $p^{n+1}$  to previous values:  $\mathbf{u}^{n+1} = \mathbf{u}^n$ ,  $p^{n+1} = p^n$ ;
    Solve momentum equation for  $\mathbf{u}^{n+1}$  prediction :  $-\mu \nabla^2 \mathbf{u}^{n+1} + \rho_{el} \nabla \phi = -\nabla p^{n+1}$ ;
    while Pressure correction do
      Solve pressure correction equation:  $\nabla \cdot \left[ \left( \frac{1}{A} \right)_f \nabla p^{n+1} \right] = \nabla \cdot \left( \frac{\mathbf{H}}{A} \right)_f$ ;
      Correct flux:  $\mathbf{U}^{n+1} = \mathbf{u}_f \cdot \mathbf{S}_f = \left( \frac{\mathbf{H}}{A} \right)_f \cdot \mathbf{S}_f - \left( \frac{1}{A} \right)_f \mathbf{S}_f \cdot \nabla_f^\perp p^{n+1}$ ;
      Correct velocity:  $\mathbf{u}^{n+1} = \frac{\mathbf{H}}{A} - \frac{1}{A} \nabla p^{n+1}$ 
    end
    while Non-orthogonal correction do
      Solve Poisson's:  $-\varepsilon \nabla^2 \phi_f = \rho_{el}$ ;
    end
    for All  $c_{i,f}$ 's do
      while Non-orthogonal correction do
        Solve Nernst-Planck:  $\frac{\partial c_{i,f}}{\partial t} + \nabla \cdot \left( \mathbf{u} c_{i,f} - D_i \left( \nabla c_{i,f} + \frac{F}{RT} z_i c_{i,f} \nabla \phi_f \right) \right) = 0$ ;
      end
    end
  end
  for All solid regions do
    while Non-orthogonal correction do
      Solve Poisson's:  $-\varepsilon \nabla^2 \phi_s = \rho_{el}$ ;
    end
    for All  $c_{i,s}$ 's do
      while Non-orthogonal correction do
        Solve Diffusion equation:  $\frac{\partial c_{i,s}}{\partial t} - D_i \nabla^2 c_{i,s} = 0$ ;
      end
    end
  end
end
Increment time:  $t \rightarrow t + \Delta t$ ;
end

```

tion:

$$\beta_s X_s \leftrightarrow \beta_f X_f. \tag{65}$$

The general reactive conditions eq. (51) in this case reduces to the following condition along Γ :

$$\mathbf{j}_f|_{\Gamma_f} \cdot \mathbf{n} = \mathbf{j}_s|_{\Gamma_s} \cdot \mathbf{n} = r(c_f|_{\Gamma_f}, c_s|_{\Gamma_s}), \tag{66}$$

where we denote here Γ_s to be evaluation at the Ω_s side of Γ and \mathbf{n} the unit normal of Γ facing into Ω_f . Note the reaction rate r is kept general and not necessarily linear. Equation 66 is solved iteratively using the Newton–Raphson method, linearised about the previous solution $\mathbf{c}^N = (c_s^N \ c_f^N)^\top$. Here the previous solution could be the solution at the previous time step (if an explicit time stepping is chosen) or at the previous internal iteration (for fully implicit time stepping). This allows us to rewrite Eq. (66) into two decoupled effective Robin conditions for c_s^{N+1} and c_f^{N+1} that we solve

separately. The coupling between the two sides of the interface conditions (and therefore the two domains) is achieved through the internal iterations but, for stiff reactions, additional sub-looping to update both boundary values whenever one of the two domains is solved for. In Algorithm 3 we detail the *mappedChemicalKinetics* pseudo-code to solve the effective Robin conditions. As both resulting effective Robin conditions are solved in the same manner we only write here solving of the condition for c_f .

More details about the linearisation can be found in Appendix A. Alongside the reactive conditions we have also implemented a number of simpler conditions, such as the continuity of total fluxes, continuity of value or continuity of derivatives, often used within applications. Just as with the non-linear reactive condition, we rewrite all conditions into effective Robin conditions.

Algorithm 3: mappedChemicalKinetics

```

Set tolerance: tol;
Set max Newton iterations:  $N_{\max}$ ;
Reset iterator:  $N = 0$ ;
while  $\|c_f^N - c_f^{N+1}\| > \text{tol} \cap N \leq N_{\max}$  do
  Store previous iteration:  $c_f^N = c_f^{N+1}$ ;
  Extract the face and internal values of the neighbouring field (solid side);
  Update  $D_f^*$ ,  $K_f^*$  and  $F_f^*$ : see eqs. (109) to (111) ;
  Evaluate iterative effective Robin BC:  $-D_f^* \nabla_{\mathbf{n}} c_f^{N+1} = -K_f^* c_f^{N+1} - F_f^*$ ;
  Apply the boundary condition to obtain the new value:  $c_f^{N+1}$ ;
  if Stiff then
    | Call mappedChemicalKinetics BC on the solid side;
  end
  Increment iterator:  $N \rightarrow N + 1$ 
end

```

6 Numerical examples

Here we present four numerical examples of electrokinetic flows. To verify accuracy of results we compare with the spectral MATLAB[®] toolbox *Chebfun* [9] with machine-precision accuracy. The first case verifies accuracy of the flow description given a single ion species using a pressure-driven infinite ion channel similar to [4]. Next we verify accuracy when considering multiple ion species. Afterwards we verify the implemented reactive interface conditions counterpart to §4.2.1. The final case displays the capabilities of our solver(s), simulating ionic transport within a randomised solid–fluid porous medium.

To show spatial convergence of OpenFOAM[®] results we use the following normalised L^2 error point norm. We use here the dummy variables v and v_{cref} to denote OpenFOAM[®] and *Chebfun* results respectively as an example:

$$L^2\text{-norm error} = \frac{\|v - v_{\text{cref}}\|_2}{\|v_{\text{cref}}\|_2}. \quad (67)$$

6.1 SPNP in an infinite channel

To verify *pnpFoam* and *pnpMultiFoam* for a single ion species take Ω , of length L , with two boundaries Γ_t and Γ_b denoting the top and bottom channel walls respectively. Take boundaries Γ_{in} and Γ_{out} denoting the inlet and outlet of fluid in Ω . To mathematically describe this channel as infinite in length we take periodic conditions on Γ_{in} and Γ_{out} .

Take a monovalent ionic fluid, i.e a single ion species c with valency $z = 1$. Enforce a fixed external force $\mathbf{J} = \begin{pmatrix} -\mu/H^2 \\ 0 \end{pmatrix}$ to induce transport across the channel length. To allow electro-migration of c between the two channel walls,

fix $\phi = 0.1$ along Γ_t and $\phi = 0$ on Γ_b . For c apply no-flux along Γ_t and Γ_b , denoting \mathbf{n} to be the outward unit normal of either Γ_t or Γ_b . We consider the channel at steady state and, by assuming L is sufficiently large, take all derivatives in x to be zero, i.e. $\frac{\partial a}{\partial x} = 0$ for $a \in \{\mathbf{u}, \phi, c, p\}$. Just like Poiseuille flow we assume a uni-directional velocity, such that $\mathbf{u} = \begin{pmatrix} u \\ 0 \end{pmatrix}$. These assumptions in turn produce a one-dimensional reduce system of equations in y .

The SPNP system with a single ionic species in an infinite channel is considered with an added pressure-driven driving force \mathbf{J} :

$$\mu \nabla^2 \mathbf{u} - \mathbf{J} - \nabla p - Fc \nabla \phi = 0, \quad \mathbf{x} \in \Omega, \quad (68)$$

$$\nabla \cdot \mathbf{u} = 0, \quad \mathbf{x} \in \Omega, \quad (69)$$

$$\epsilon \nabla^2 \phi + Fc = 0, \quad \mathbf{x} \in \Omega, \quad (70)$$

$$\nabla \cdot \left[-D \left(\nabla c + \frac{F}{RT} c \nabla \phi \right) + \mathbf{u} c \right] = 0, \quad \mathbf{x} \in \Omega, \quad (71)$$

$$\mathbf{u} = \mathbf{0}, \quad \phi = 0.05, \quad \mathbf{x} \in \Gamma_t \quad (72)$$

$$\mathbf{u} = \mathbf{0}, \quad \phi = 0, \quad \mathbf{x} \in \Gamma_b, \quad (73)$$

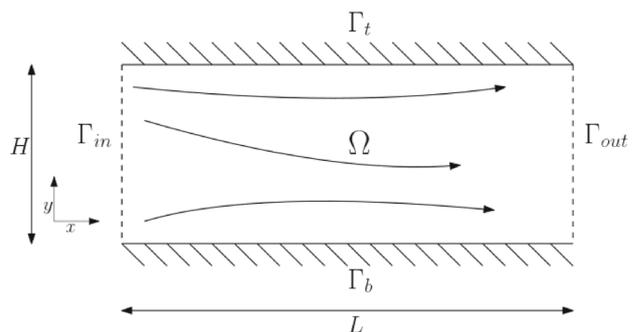
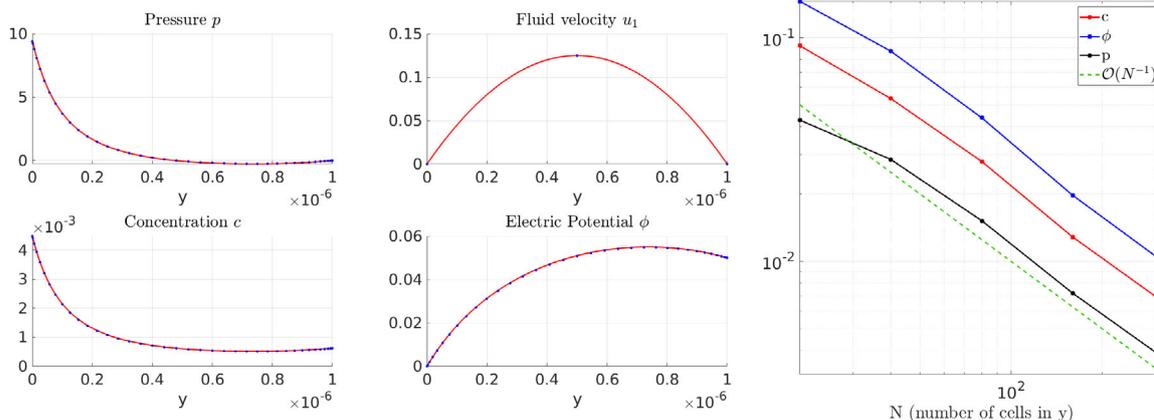


Fig. 3 Graphical representation of the infinite ion channel with channel width H

Table 1 Dimensionless numbers, domain and transport properties for the SPNP equation in the infinite channel. We denote $\epsilon_0 = 8.85 \times 10^{-12}$ to be the permittivity of vacuum

Symbol	μ	H	ϵ	\mathcal{N}	Pe	\mathcal{L}^2	\mathcal{P}
Value	10	1×10^{-6}	$40\epsilon_0$	1.95	1.25×10^5	0.09	3.85×10^{-6}



(a) Comparison of *Chebfun* (blue dots) and *pnpFoam* (red line) of the pressure, fluid velocity, ion concentration, and electric potential respectively, along the channel width i.e., in y . (b) L^2 error norm convergence plot of *pnpFoam* results, showing spatial convergence of $\mathcal{O}(N^{-1})$

Fig. 4 **a** Comparison of *Chebfun* (blue dots) and *pnpFoam* (red line) of the pressure, fluid velocity, ion concentration, and electric potential respectively, along the channel width i.e., in y . **b** L^2 error norm convergence plot of *pnpFoam* results, showing spatial convergence of $\mathcal{O}(N^{-1})$

$$-D \left(\nabla_{\mathbf{n}} c + \frac{F}{RT} c \nabla_{\mathbf{n}} \phi \right) + c \mathbf{u} \cdot \mathbf{n} = 0, \quad \mathbf{x} \in \Gamma_b \cup \Gamma_t, \quad \frac{dc}{dy} + \frac{F}{RT} c \frac{d\phi}{dy} = 0, \quad \mathbf{x} \in \Gamma_b \cup \Gamma_t, \quad (74)$$

$$a \Big|_{\Gamma_{in}} = a \Big|_{\Gamma_{out}}, \quad \frac{\partial a}{\partial x} \Big|_{\Gamma_{in}} = \frac{\partial a}{\partial x} \Big|_{\Gamma_{out}}, \quad a \in \{c, \mathbf{u}, \phi, p\}. \quad (75)$$

The complete system is solved in OpenFOAM® in a periodic channel, and compared with the equivalent one-dimensional model. By applying the assumptions of unidirectional flow and zero derivatives in x we obtain a one-dimensional reduced set of equations in y :

$$\mu \frac{d^2 u_1}{dy^2} = -\frac{\mu}{H^2}, \quad \mathbf{x} \in \Omega, \quad (76)$$

$$0 = \frac{dp}{dy} + Fc \frac{d\phi}{dy}, \quad \mathbf{x} \in \Omega, \quad (77)$$

$$\epsilon \frac{d^2 \phi}{dy^2} = -Fc, \quad \mathbf{x} \in \Omega, \quad (78)$$

$$0 = \frac{d^2 c}{dy^2} + \frac{F}{RT} \left(\frac{dc}{dy} \frac{d\phi}{dy} + c \frac{d^2 \phi}{dy^2} \right), \quad \mathbf{x} \in \Omega, \quad (79)$$

$$u_1 = 0, \quad \phi = 0.05, \quad \mathbf{x} \in \Gamma_t, \quad (80)$$

$$u_1 = 0, \quad \phi = 0, \quad \mathbf{x} \in \Gamma_b, \quad (81)$$

where we excluded the incompressibility condition as this is trivially satisfied.

To compare results between *Chebfun* and our single ionic fluid solver *pnpFoam* we use the following dimensionless numbers, geometrical parameters and transport properties, according to Table 1.

When running *pnpFoam* we solve it in steady state and take a large number of PIMPLE corrector iterations to insure convergence of the coupled system. The mesh used in OpenFOAM® is up to $N = 420$ cells (100 in x , 320 in y). Comparison of results and the L^2 error norm convergence are shown in fig. 4. We observe an accumulation of c along Γ_b , causing a large pressure gradient to form. The velocity profile follows a parabolic arc, as expected, whilst we see a non-linear profile for ϕ . Overall we find good agreement, with linear order spatial convergence $\mathcal{O}(N^{-1})$.

6.2 Multi-component ionic fluid

To verify *pnpFoam* and *pnpMultiFoam* for a multi-component fluid we take two species c_1 and c_2 with opposite valencies $z_1 = -z_2 = 1$ over the domain $\Omega \in [0, L]$ where $L = 1 \times 10^{-6}$. We set a zero-flux BC, i.e., $j(c_i) = 0$ at

Table 2 Dimensionless numbers and parameters for the multi-component ionic fluid testcase

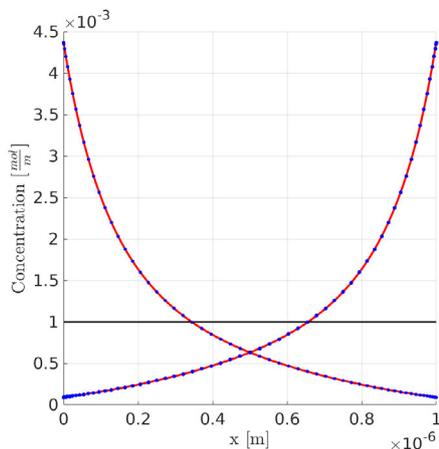
Symbol	\mathcal{N}	Pe	\mathcal{L}	\mathcal{P}
Value	3.868	1×10^{-3}	9×10^{-3}	0

$x = 0, L$ for both species and fix $\phi = 0.1$ at $x = L$ and $\phi = 0$ at $x = 0$. We fix here a constant velocity $\mathbf{u} = 0.001$ for simplicity and we by-pass the solution of the Stokes system (*frozenFlow* flag). Concentrations and potential are initially set to $c_i = 1 \times 10^{-3}$ and $\phi = 0.05(1 - \cos \frac{\pi x}{L})$.

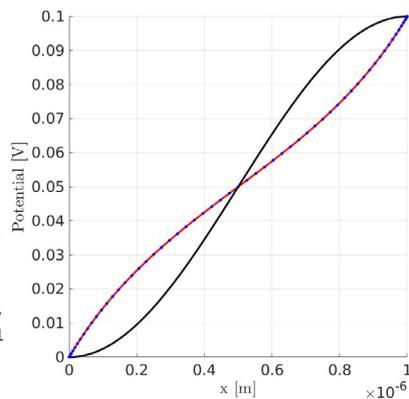
We consider here the dimensionless number values in Table 2, indicating electrostatic forces dominate. From \mathcal{L} we find the Debye length as approximately $\lambda_D = 308\text{nm}$. Since we are fixing \mathbf{u} and neglecting Stokes we have $\mathcal{P} = 0$.

When running *pnpFoam* we use a time step $\Delta t = 10^{-7}\text{s}$, end time $t = 1 \times 10^{-5}\text{s}$, third-order implicit time scheme (*backward* keyword) and a mesh of up to $N = 1000$ cells. Results and L^2 norm convergence plots are depicted in Sect. 6.3 where we find good agreement of results and linear spatial convergence of all fields.

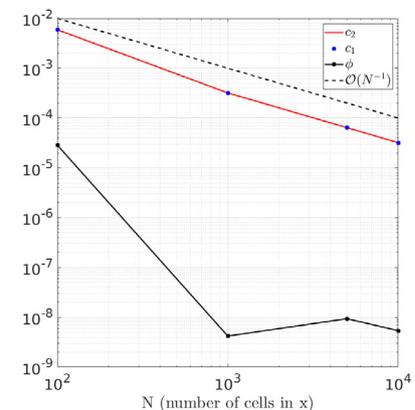
We see the ions are transported to the outer walls due to the high electric potential gradient. Most of the ions then accumulate within λ_D from the walls forming two overlapping EDLs. Non-linear behaviour between c_i and ϕ is observed through the slight shift in ϕ 's profile due to the clustering of ions at the walls.



(a) Results of c_1 (left curve) and c_2 (right) at $t = 1 \times 10^{-5}\text{s}$. Blue and red denote *Chebfun* and *pnpFoam* results respectively. Black is the initial condition.



(b) Results of ϕ for *Chebfun* (blue) and *pnpFoam* (red) at $t = 1 \times 10^{-5}\text{s}$, with initial condition in black.



(c) L^2 point error norm convergence plot, showing $\mathcal{O}(N^{-1})$ spatial convergence of *pnpFoam*.

Fig. 5 **a** Results of c_1 (left curve) and c_2 (right) at $t = 1 \times 10^{-5}\text{s}$. Blue and red denote *Chebfun* and *pnpFoam* results respectively. Black is the initial condition. **b** Results of ϕ for *Chebfun* (blue) and *pnpFoam* (red)

6.3 Reactive interface

Here we verify the accuracy of *mappedChemicalKinetics*, the numerical counterpart to the conditions found in §4.2.1. Consider the domain $\Omega = \Omega_s \cup \Omega_f = [-1, 1]$ split into fluid $\Omega_f \in [-1, 0]$ and solid $\Omega_s \in [0, 1]$, each containing the species c_f and c_s respectively. Both species are restricted to their respective sub-domain, i.e. $c_s = 0$ in Ω_f and vice versa. Consider the following elementary reaction at the $x = 0$ interface: Figure. 5



As c_s and c_f are restricted to Ω_s and Ω_f respectively the reactive conditions of Eqs. (51) and (52) become

$$j(c_f) = j(c_s), \quad x = 0, \tag{84}$$

$$j(c_f) = k_f c_s - k_r c_f, \quad x = 0, \tag{85}$$

where we have used the linear *rate law* to model the reaction rate. Note c_f and c_s in Eqs. (84) and (85) are evaluated on Ω_f and Ω_s 's side of $x = 0$ respectively. We set both species as uncharged, i.e. $z_i = 0$, to remove electrostatic effects and $u = 1$ in Ω_f to ignore Stokes. We initially set $c_s = 0$ in Ω_s and $c_f = e^{-200(x+0.5)^2}$ in Ω_f so that a clear increase in c_s can be seen.

Looking at the dimensionless numbers for the problem in Table 3 we can notice here that advection dominates. We also

at $t = 1 \times 10^{-5}\text{s}$, with initial condition in black. **c** L^2 point error norm convergence plot, showing $\mathcal{O}(N^{-1})$ spatial convergence of *pnpFoam*

Table 3 Table of dimensionless values

Symbol	\mathcal{N}	Pe	\mathcal{L}	\mathcal{P}	$Da_{II,f}$	$Da_{II,r}$
Value	0	1	0	0	10	100

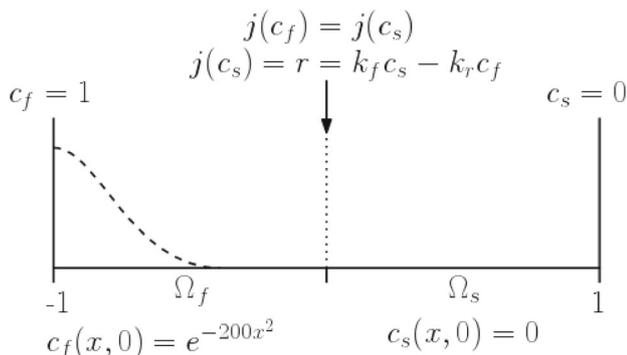


Fig. 6 Graphical representation of domain Ω containing a solid and fluid with zero flux outer boundaries and reactive flux conditions at the shared interface

include two more numbers from the reactive interface conditions at $x = 0$. These are $Da_{II,f}$ and $Da_{II,r}$, the Damköhler numbers given as the ratio between the diffusion rate and reaction rates, forward and reverse, respectively.

When running *pnpMultiFoam* to show qualitative comparison we use a time step $\Delta t = 10^{-3}$ s, second-order implicit time scheme *backwards* and mesh discretisation of $N = 1000$ cells (500 in Ω_f , 500 in Ω_s). To show convergence of results and compute the L^2 error point norm we move to the *steadyState* time scheme to compare steady states. Results of the transient case are shown on the left in Fig. 7 where we find good agreement. First an initial diffusion and advection of c_f is seen towards the interface, once c_f reaches $x = 0$ it reacts to form c_s in Ω_s , where afterwards c_s diffuses through Ω_s . After $t = 15$ s a steady state is reached implying chemical equilibrium of the reaction. As for the L^2 error norm of the steady case, seen in Fig. 7, we find a second-order convergence of *mappedChemicalKinetics* to *Chebfun*. This is because the interface conditions are linear and therefore exactly approximated by the linear approximation.

6.4 Random porous REV

To demonstrate the capabilities of our solvers, we consider a randomly generated porous solid–fluid domain $\Omega = \Omega_s \cup \Omega_f$, see Fig. 8. The random generation is done by a truncated Gaussian random field [12, 24], sampled at the mesh points and categorised into two bins, denoting solid and fluid cells, by a threshold. Raising or lowering this threshold alters the porosity of the domain.

To have Ω be a microscopic representative elementary volume (REV) of a much larger macroscopic porous medium,

we apply periodic conditions along the outer boundaries $\Gamma_{ext} = \Gamma_{in} \cup \Gamma_{out} \cup \Gamma_t \cup \Gamma_b$. To generate movement, we fix a jump in ϕ between $\Gamma_{in,s}$ and $\Gamma_{out,s}$, where the subscript s denotes the section of Γ_{in} neighbouring Ω_s . This is to mimic a fixed applied potential difference across the macroscopic medium, such as an applied voltage across a battery cell. We consider two ion species c_1 and c_2 with opposite valencies $z_1 = -z_2 = -1$. The height and length of the region is set as $H = 1 \times 10^{-4}$ m.

In Ω_f we start with a uniform concentration of c_2 , the same is done for Ω_s with c_1 . Fluid is initially taken at rest. Along Γ_{sf} we set no flux for c_1 and flux continuity for c_2 . For ϕ we assume continuity of the electric displacement $\mathbf{D} = \varepsilon \mathbf{E}$ along Γ_{sf} . In Table 4 we list the dimensionless number values of the case.

The system is initialised with constant c_1 and c_2 in Ω_f and Ω_s respectively. All fields are periodic on the outer boundaries $\Gamma_{ext} = \Gamma_{in} \cup \Gamma_{out} \cup \Gamma_t \cup \Gamma_b$, apart from ϕ where we introduce a quasi-periodic condition with a fixed jump between $\Gamma_{in,s}$ and $\Gamma_{out,s}$, denoting the parts of Γ_{in} and Γ_{out} neighbouring Ω_s . We apply no flux and flux continuity for c_1 and c_s along Γ_{sf} respectively. For ϕ we assume continuity of the electric displacement $\mathbf{D} = \varepsilon \mathbf{E}$ along Γ_{sf} :

$$c_1(\mathbf{x}, 0) = 0, \quad c_2(\mathbf{x}, 0) = 1 \times 10^{-8}, \quad \phi(\mathbf{x}, 0) = 0, \quad \mathbf{x} \in \Omega_s; \tag{86}$$

$$c_1(\mathbf{x}, 0) = 1 \times 10^{-8}, \quad c_2(\mathbf{x}, 0) = 0, \quad \phi(\mathbf{x}, 0) = 0, \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{0}, \quad p(\mathbf{x}, 0) = 0, \quad \mathbf{x} \in \Omega_f; \tag{87}$$

$$\left[\nabla_{\mathbf{n}} a \right]_{\Gamma_{out,s}}^{\Gamma_{in,s}} = 0, \quad \left[a \right]_{\Gamma_{out,s}}^{\Gamma_{in,s}} = 0, \\ \left[\nabla_{\mathbf{n}} \phi \right]_{\Gamma_{out,s}}^{\Gamma_{in,s}} = 0, \quad \left[\phi \right]_{\Gamma_{out,s}}^{\Gamma_{in,s}} = 1 \times 10^{-2}, \quad a \in \{c_1, c_2\}; \tag{88}$$

$$\left[\nabla_{\mathbf{n}} a \right]_{\Gamma_{out,f}}^{\Gamma_{in,f}} = 0, \quad \left[a \right]_{\Gamma_{out,f}}^{\Gamma_{in,f}} = 0, \quad a \in \{c_1, c_2, \phi, p, \mathbf{u}\}; \tag{89}$$

$$\left[\nabla_{\mathbf{n}} a \right]_{\Gamma_b}^{\Gamma_t} = 0, \quad \left[a \right]_{\Gamma_b}^{\Gamma_t} = 0, \quad a \in \{c_1, c_2, \phi, p, \mathbf{u}\}; \tag{90}$$

$$\mathbf{j}_1 \cdot \mathbf{n} = 0, \quad \left[\mathbf{j}_2 \cdot \mathbf{n} \right]_{\Gamma_{sf,s}}^{\Gamma_{sf,f}} = 0, \quad \nabla p \cdot \mathbf{n} = 0, \quad \mathbf{u} \cdot \mathbf{n} = 0, \\ \left[\varepsilon \nabla_{\mathbf{n}} \phi \right]_{\Gamma_{sf,s}}^{\Gamma_{sf,f}} = 0, \quad \mathbf{x} \in \Gamma_{sf}. \tag{91}$$

To show consistency in the results we run four simulations in total. This is done for two randomly generated meshes, each discretised over two levels of refinement for $N = 1 \times$

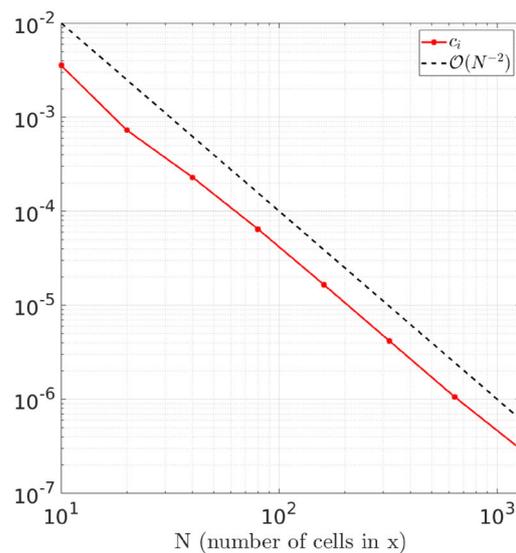
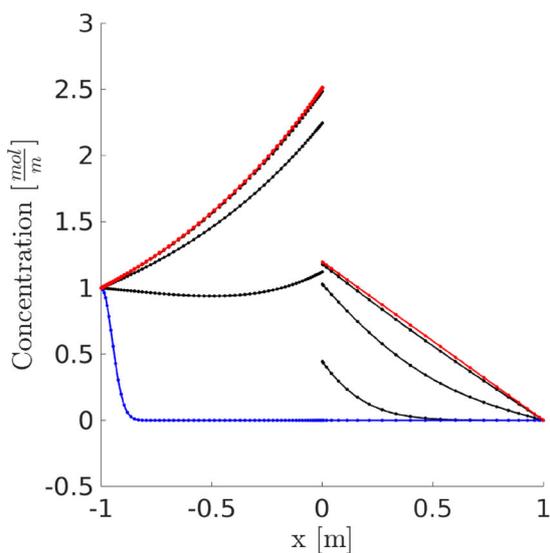


Fig. 7 Left: Molar concentration results of c_f in fluid (dashed for *pnpMultiFoam*, markers for *Chebfun*) and c_s in solid (solid for *pnpMultiFoam*, markers for *Chebfun*). Blue for initial concentrations, black

at intermediate times, red for final time (i.e steady state), Right: L^2 norm convergence plot between *Chebfun* and mappedChemicalKinetics/*pnpMultiFoam*

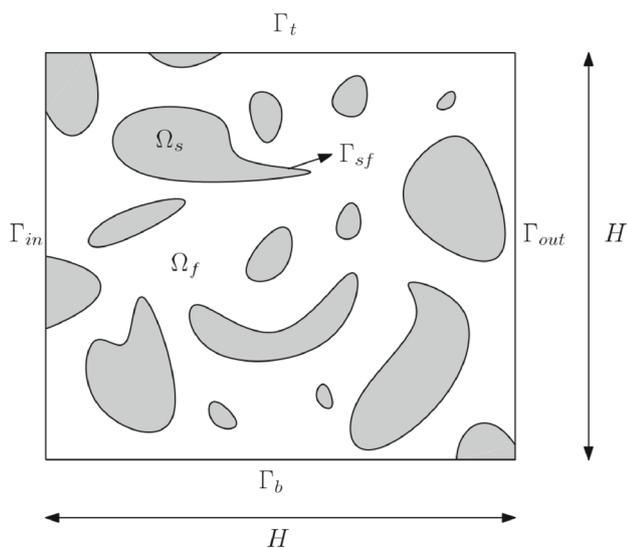


Fig. 8 Visual representation of the random porous domain containing solid Ω_s (grey) and fluid Ω_f (white) with shared interface Γ_{sf}

10^4 and $N = 4 \times 10^4$ cells. We will use letters A and B to differentiate between the random generations and subscripts 1 and 2 for the levels of refinement. E.g. A_2 denotes the first random mesh, refined using 4×10^4 cells. All runs use a time step $\Delta t = 5 \times 10^{-4}$ and second-order implicit time scheme *backwards*. To observe the advective and electrostatic effects on our ions we define the following velocities in Ω_f :

$$\mathbf{v}_i = \mathbf{u} - \frac{D_i z_i F}{RT} \nabla \phi, \tag{92}$$

Table 4 Dimensionless numbers for the random porous media testcase

Symbol	\mathcal{N}	Pe	\mathcal{L}	\mathcal{P}
Value	0.4	9.65×10^{-4}	0.95	1

such that the continuity Eq. (5) in Ω_f may be written as

$$\frac{\partial c_i}{\partial t} + \nabla \cdot (\mathbf{v}_i c_i - D_i \nabla c_i) = 0. \tag{93}$$

Figure 9 shows the results of the first random realisation with a finer mesh ($N = 4 \times 10^4$), where we observe a gradient in ϕ formed between regions of Ω_s connected at Γ_{in} and Γ_{out} . This is primarily due to our jump condition for ϕ applied along $\Gamma_{in,s}$ and $\Gamma_{out,s}$. Whilst movement of our ion species have an effect on this gradient, for the most part it remains steady. The applied electrical driving force causes the uniform concentration c_1 in Ω_f to accumulate around the regions of Ω_s connected at Γ_{out} , where a higher electric potential is present. Conversely, around Ω_s connected at Γ_{in} we find a much lower, but not zero, concentration of c_1 . A similar observation is found with c_2 , once diffused out of Ω_s , accumulating where there is a lower potential. As with c_1 , at the region of Ω_s connected at Γ_{out} there is a much lower, but again not zero, concentration of c_2 . We observe therefore the formation of EDL's, see §3, around these connected solids. Given more time to evolve these EDL's would become further apparent, with more of c_2 diffusing out of Ω_s . As for \mathbf{v}_1 and \mathbf{v}_2 we find their magnitudes $\|\mathbf{v}_1\|$ and $\|\mathbf{v}_2\|$ are highest

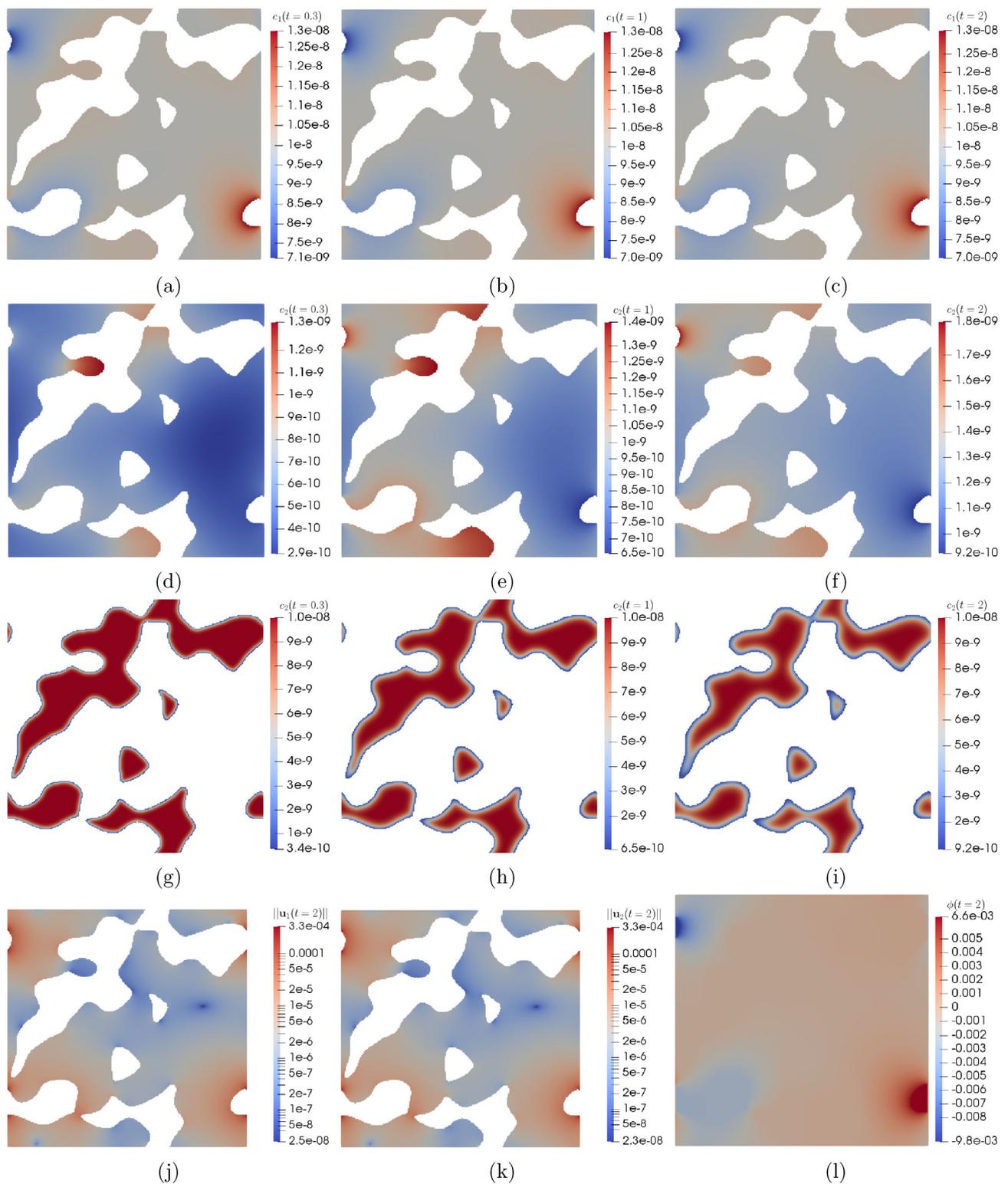


Fig. 9 Results of the first random solid–fluid geometry, discretised using 4×10^4 cells. **a–f** Profiles of c_1 and c_2 at $t = 0.3, 1, 2$ s within Ω_f . **g–i** Profiles of c_2 at $t = 0.3, 1, 2$ s within Ω_s . **j–l** Profiles of $\|v_1\|$, $\|v_2\|$ and ϕ at $t = 2$ s

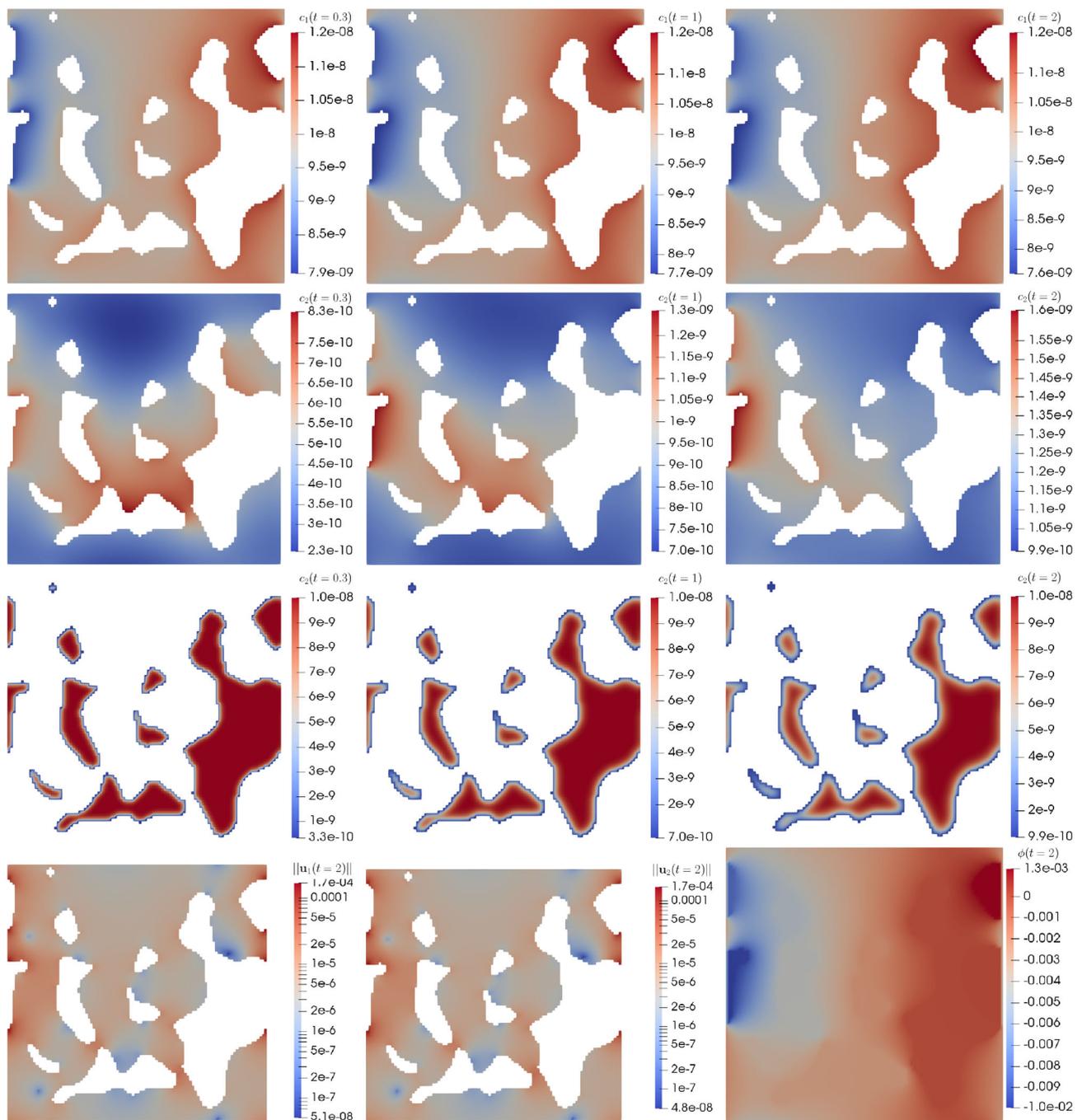


Fig. 10 Results of the first random solid–fluid geometry, discretised using $N = 1 \times 10^4$ cells. Top two rows: Profiles of c_1 and c_2 at $t = 0.3, 1, 2$ s within Ω_f . Third row: Profiles of c_2 at $t = 0.3, 1, 2$ s within Ω_s . Bottom row: Profiles of $\|u_1\|$, $\|u_2\|$ and ϕ at $t = 2$ s

within these EDL's located at the connected solids due to higher gradients in ϕ .

We can compare these results with coarse mesh ($N = 1 \times 10^4$) results as well as with other random generation seed. Results of first realisation with a coarse mesh, and of a coarse and fine simulation of a second random realisation are reported respectively in Figs. 10, 11, and 12.

7 Conclusions

In this paper we present the development of open-source solvers in OpenFOAM[®] capable of simulating microscopic electrokinetic flows of dilute ionic fluids. The underlying system, known as Stokes-Poisson-Nernst-Planck, is thoroughly reviewed where we discuss the assumptions on the fluid properties and limitations these bring. We later apply

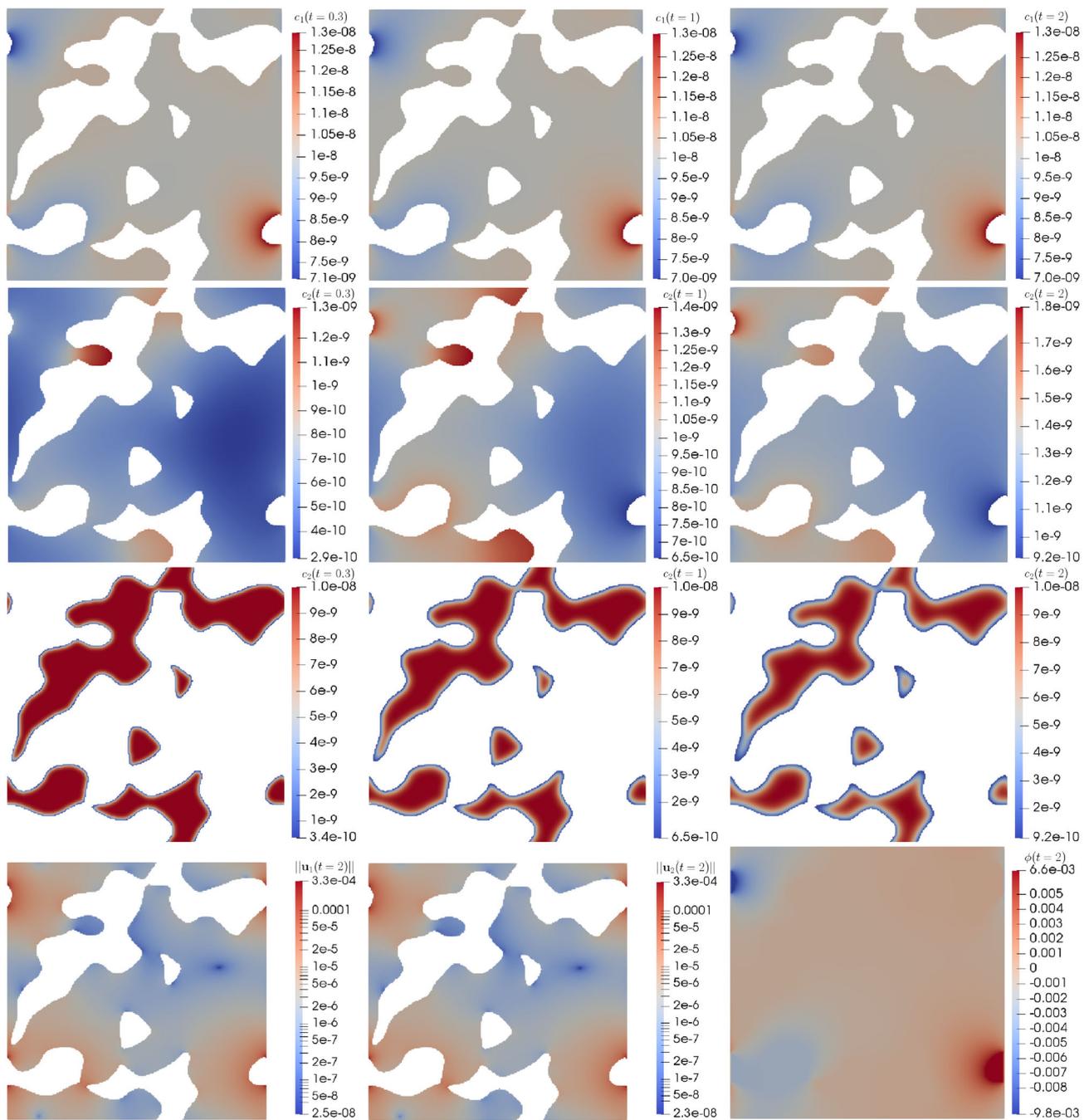


Fig. 11 Results of the second random solid–fluid geometry, discretised using $N = 1 \times 10^4$ cells. Top two rows: Profiles of c_1 and c_2 at $t = 0.3, 1, 2$ s within Ω_f . Third row: Profiles of c_2 at $t = 0.3, 1, 2$ s within Ω_s . Bottom row: Profiles of $\|\mathbf{u}_1\|, \|\mathbf{u}_2\|$ and ϕ at $t = 2$ s

dimensional analysis to characterise the effects of dominating physical forces. This analysis is later used to give better understanding to a common model reduction, known as the electro-neutrality assumption, as a result of such analysis.

Many real-world applications of these flows involve some form of reactions between involved ionic species at the interface between fluid and solid. As such, we formulate a fully general reaction model capable of describing said hetero-

geneous reactions as a balance of fluxes across all reacting species.

To verify flow descriptions obtained from our solvers are accurate, we compare against highly accurate solutions obtained, for simplified cases, with the Matlab toolbox *Chebfun* for a number of cases. With each case we find good agreement, showing spatial grid convergence of the results. We later use our solvers across a randomly generated solid–

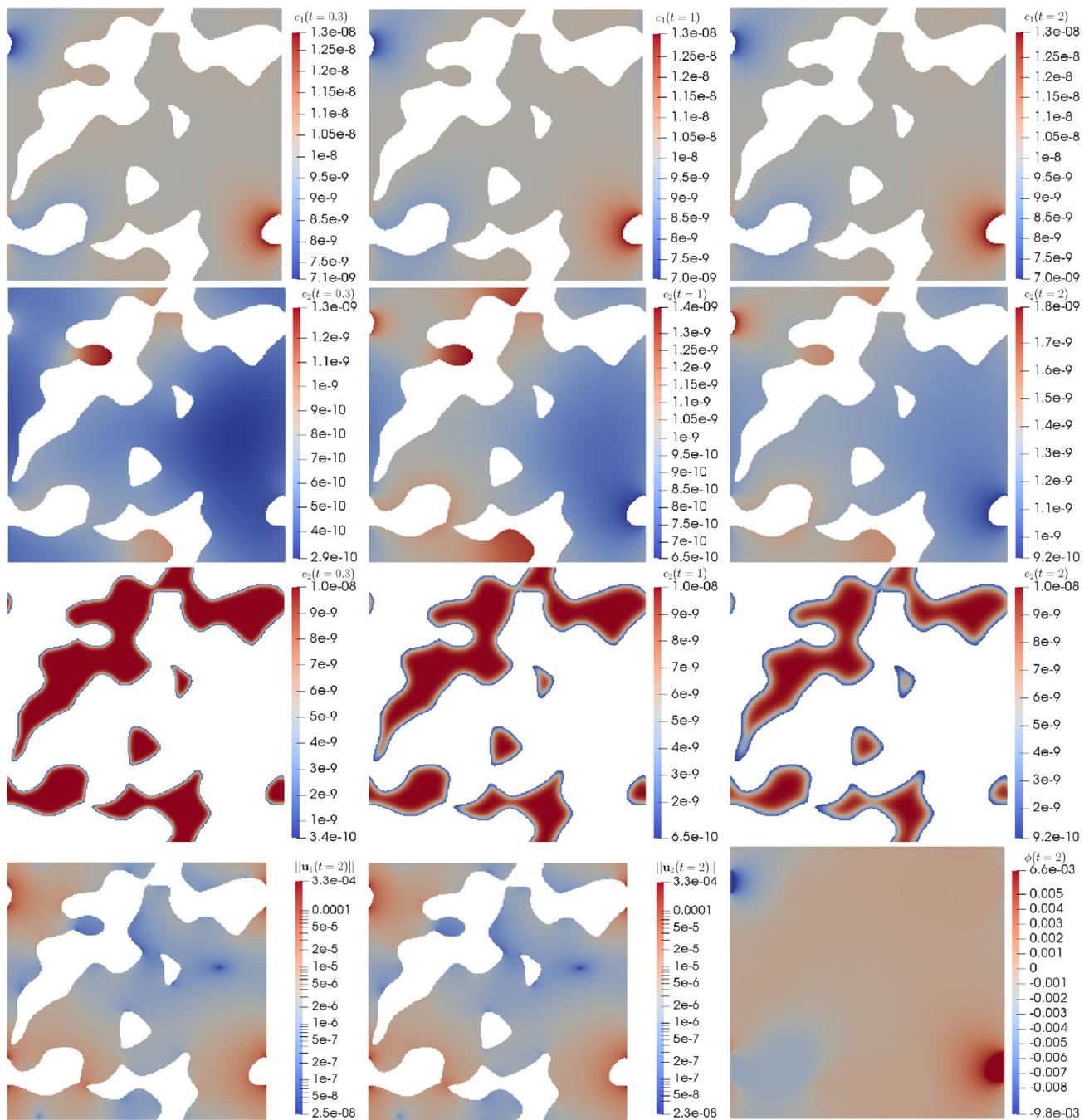


Fig. 12 Results of the second random solid–fluid geometry, discretised using $N = 4 \times 10^4$ cells. Top two rows: Profiles of c_1 and c_2 at $t = 0.3, 1, 2$ s within Ω_f . Third row: Profiles of c_2 at $t = 0.3, 1, 2$ s within Ω_s . Bottom row: Profiles of $\|\mathbf{u}_1\|$, $\|\mathbf{u}_2\|$ and ϕ at $t = 2$ s

fluid porous cell, similar to a representative elementary volume (REV) used in homogenisation theory, to show the solvers ability in handling complex micro-structures.

In the future we plan on constructing full physically realistic cases, such as modelling the electrokinetic flow through a porous battery half-cell. Furthermore we later plan to apply uncertainty quantification in conjunction with these

randomly generated porous cells, quantifying the effects on the flow for varying geometrical properties.

Declarations

Conflict of interest The authors declare that they have no conflicts of interest. MI and RB have been supported by the University of Nottingham Impact Acceleration Award.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

A. Quasi-coupled Newton iterative effective reactive Robin conditions

A.1. General non-linear reaction rate for restricted binary reaction

Let there be the following reversible binary reaction between species c_s and c_f along the interface Γ shared by Ω_s and Ω_f ,



and consider each species is restricted to their respective sub-domains:

$$c_s = \begin{cases} 0 & \mathbf{x} \in \Omega_f \\ c_s & \mathbf{x} \in \Omega_s \end{cases}, \quad c_f = \begin{cases} c_f & \mathbf{x} \in \Omega_f \\ 0 & \mathbf{x} \in \Omega_s \end{cases}. \tag{95}$$

In these circumstances we have the following condition along Γ with reaction rate r :

$$\mathbf{j}_s|_{\Gamma_s} \cdot \mathbf{n} = r(c_s|_{\Gamma_s}, c_f|_{\Gamma_f}) = -\mathbf{j}_f|_{\Gamma_f} \cdot \mathbf{n}'. \tag{96}$$

We denote here \mathbf{n} to be the unit normal along Γ facing outwards of Ω_s and $\mathbf{n}' = -\mathbf{n}$. Consider the reaction rate r is a non-linear function of c_s and c_f . To obtain effective Robin conditions of Eq. (96) for c_s and c_f we employ the Newton–Raphson method. First, we take the vector form of Eq. (96) with all terms moved to the left. To simplify writing, for now we omit what side of Γ each term is evaluated on:

$$\begin{aligned} \mathbf{J}(c_s, c_f) &= \begin{pmatrix} J_1 \\ J_2 \end{pmatrix} \\ &\equiv \begin{pmatrix} -D_s \nabla_{\mathbf{n}} c_s - r(c_s, c_f) \\ (\mathbf{u}_f \cdot \mathbf{n}') c_f - D_f \nabla_{\mathbf{n}'} c_f - \frac{z_f D_f F}{RT} c_f \nabla_{\mathbf{n}'} \phi_f + r(c_s, c_f) \end{pmatrix} \\ &= \mathbf{0}. \end{aligned} \tag{97}$$

Assume we have a current approximate solution $\mathbf{c}^N = (c_s^N \ c_f^N)^T$. Linearising \mathbf{J} around \mathbf{c}^N then gives

$$\mathbf{J}(\mathbf{c}^N) + \frac{\partial \mathbf{J}}{\partial \mathbf{c}} \Big|_{\mathbf{c}^N} (\mathbf{c} - \mathbf{c}^N) = \mathbf{0}, \tag{98}$$

whereby setting $\mathbf{c} = \mathbf{c}^{N+1}$ is the next iterative solution. We denote here $\frac{\partial \mathbf{J}}{\partial \mathbf{c}}$ to be the Jacobian matrix of \mathbf{J} . To determine the Jacobian we define the directional derivative for some function $G(c)$ along the direction $\delta c = c - c^N$ as:

$$\frac{\partial G}{\partial c}(\delta c) = \lim_{\varepsilon \rightarrow 0} \frac{G(c + \varepsilon \delta c) - G(c)}{\varepsilon}. \tag{99}$$

Using this definition of the directional derivative we can then determine the Jacobian within Eq. (98). Since the method is similar for all components of \mathbf{J} , we will focus on the first element J_1 . Taking the directional derivative of J_1 with respect to c_s along δc_s at $\mathbf{c} = \mathbf{c}^N$ gives:

$$\begin{aligned} \frac{\partial J_1}{\partial c_s} \Big|_{\mathbf{c}^N}(\delta c_s) &= \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} \left(-D_s \nabla_{\mathbf{n}}(c_s^N + \varepsilon \delta c_s) \right. \\ &\quad \left. - r(c_s^N + \varepsilon \delta c_s, c_f^N) + D_s \nabla_{\mathbf{n}} c_s^N + r(c_s^N, c_f^N) \right). \end{aligned} \tag{100}$$

Taking the following Taylor expansions about $\varepsilon = 0$ of r and the concentration gradient,

$$\begin{aligned} \nabla_{\mathbf{n}}(c_s^N + \varepsilon \delta c_s) &= \nabla_{\mathbf{n}} c_s^N + \varepsilon \nabla_{\mathbf{n}} \delta c_s + \mathcal{O}(\varepsilon^2), \tag{101} \\ r(c_s^N + \varepsilon \delta c_s, c_f^N) &= r(c_s^N, c_f^N) + \varepsilon \delta c_s \frac{\partial r}{\partial c_s} \Big|_{\mathbf{c}^N} + \mathcal{O}(\varepsilon^2), \end{aligned} \tag{102}$$

then substituting into Eq. (100) we obtain:

$$\frac{\partial J_1}{\partial c_s} \Big|_{\mathbf{c}^N}(\delta c_s) = -D_s \nabla_{\mathbf{n}} \delta c_s - \delta c_s \frac{\partial r}{\partial c_s} \Big|_{\mathbf{c}^N}. \tag{103}$$

The same can be done to find the directional derivative of J_1 with respect to c_f , giving:

$$\frac{\partial J_1}{\partial c_f} \Big|_{\mathbf{c}^N}(\delta c_f) = -\delta c_f \frac{\partial r}{\partial c_f} \Big|_{\mathbf{c}^N}. \tag{104}$$

Repeating the same process for J_2 of Eq. (97) and collecting all elements together and evaluating at \mathbf{c}^N gives

$$\begin{aligned} \frac{\partial \mathbf{J}}{\partial \mathbf{c}} \Big|_{\mathbf{c}^N}(\delta \mathbf{c}) &= \begin{pmatrix} -D_s \nabla_{\mathbf{n}} \delta c_s - \delta c_s \frac{\partial r}{\partial c_s} - \delta c_f \frac{\partial r}{\partial c_f} \\ \delta c_s \frac{\partial r}{\partial c_s} + (\mathbf{u}_f \cdot \mathbf{n}') \delta c_f - \frac{z_f D_f F}{RT} \delta c_f \nabla_{\mathbf{n}'} \phi_f + \delta c_f \frac{\partial r}{\partial c_f} \end{pmatrix}, \end{aligned} \tag{105}$$

where derivatives in r are evaluated at \mathbf{c}^N . We may then substitute Eq. (105) into Eq. (98) to get

$$\begin{aligned} \begin{pmatrix} -D_s \nabla_{\mathbf{n}} c_s - r - \delta c_f \frac{\partial r}{\partial c_f} - \delta c_s \frac{\partial r}{\partial c_s} \\ (\mathbf{u}_f \cdot \mathbf{n}') c_f - D_f \nabla_{\mathbf{n}'} c_f - \frac{z_f D_f F}{RT} c_f \nabla_{\mathbf{n}'} \phi_f + r + \delta c_s \frac{\partial r}{\partial c_s} + \delta c_f \frac{\partial r}{\partial c_f} \end{pmatrix} \\ = \mathbf{0}, \end{aligned} \tag{106}$$

where r and its derivatives are evaluated at \mathbf{c}^N . Re-arranging the first element of Eq. 106, using the first order approximation $\nabla_{\mathbf{n}}c_s = B_s(c_s - c_{s,c})$ where $c_{s,c}$ is the value at the cell centre of cells sharing a face with Γ , and substituting into the second element then gives:

$$\begin{aligned}
 &(\mathbf{u}_f \cdot \mathbf{n}')c_f - D_f \nabla_{\mathbf{n}'}c_f - \frac{z_f D_f F}{RT} c_f \nabla_{\mathbf{n}'}\phi_f \\
 &= -r - \delta c_f \frac{\partial r}{\partial c_f} + c_s^N \frac{\partial r}{\partial c_s} + P(c_f), \tag{107}
 \end{aligned}$$

where

$$P(c_f) = \frac{\partial r}{\partial c_s} \left(\frac{-D_s B_s c_{s,c} + r + \delta c_f \frac{\partial r}{\partial c_f} - c_s^N \frac{\partial r}{\partial c_s}}{D_s B_s + \frac{\partial r}{\partial c_s}} \right). \tag{108}$$

Setting $\mathbf{c} = \mathbf{c}^{N+1}$ to indicate the next iterative solution results in an iterative Robin condition of the form $-D_f^* \nabla_{\mathbf{n}'}c_f^{N+1} = -K_f^* c_f^{N+1} - F_f^*$, with effective coefficients dependent on the previous iteration \mathbf{c}^N :

$$D_f^* = D_f, \tag{109}$$

$$K_f^* = -\frac{\frac{\partial r}{\partial c_s} \frac{\partial r}{\partial c_f}}{D_s B_s + \frac{\partial r}{\partial c_s}} + \frac{\partial r}{\partial c_f} - \frac{z_f D_f F}{RT} \nabla_{\mathbf{n}'}\phi_f + (\mathbf{u}_f \cdot \mathbf{n}'), \tag{110}$$

$$\begin{aligned}
 F_f^* &= r - c_f^N \frac{\partial r}{\partial c_f} - c_s^N \frac{\partial r}{\partial c_s} - \frac{\partial r}{\partial c_s} \\
 &\left(\frac{-D_s B_s c_{s,c} + r - c_f^N \frac{\partial r}{\partial c_f} - c_s^N \frac{\partial r}{\partial c_s}}{D_s B_s + \frac{\partial r}{\partial c_s}} \right). \tag{111}
 \end{aligned}$$

Remember, r and its derivatives are evaluated at the previous iteration \mathbf{c}^N . Similar steps can also be applied to the accompanying iterative Robin condition for c_s of the form $-D_s^* \nabla_{\mathbf{n}}c_s^{N+1} = -K_s^* c_s^{N+1} - F_s^*$, with coefficients:

$$D_s^* = D_s, \tag{112}$$

$$K_s^* = \frac{\frac{\partial r}{\partial c_s} \frac{\partial r}{\partial c_f}}{(\mathbf{u}_f \cdot \mathbf{n}') - D_f B_f - \frac{z_f D_f F}{RT} \nabla_{\mathbf{n}'}\phi_f + \frac{\partial r}{\partial c_f}} - \frac{\partial r}{\partial c_s}, \tag{113}$$

$$\begin{aligned}
 F_s^* &= -r + c_f^N \frac{\partial r}{\partial c_f} + c_s^N \frac{\partial r}{\partial c_s} - \frac{\partial r}{\partial c_f} \\
 &\left(\frac{-D_f B_f c_{f,c} - r + c_f^N \frac{\partial r}{\partial c_f} + c_s^N \frac{\partial r}{\partial c_s}}{(\mathbf{u}_f \cdot \mathbf{n}') - D_f B_f - \frac{z_f D_f F}{RT} \nabla_{\mathbf{n}'}\phi_f + \frac{\partial r}{\partial c_f}} \right). \tag{114}
 \end{aligned}$$

A.2. General non-linear unrestricted reaction rate

Here we will formulate a set of iterative Robin conditions required to solve the reactive conditions for unrestricted species and non-linear reaction. Consider a general non-linear reaction with $i = J, \dots, K$ reactants and $i = K +$

$1, \dots, M$ products:

$$\sum_{i=J}^K v_i B_i + n e^- = \sum_{i=K+1}^M v_i B_i. \tag{115}$$

We model Eq. (115) using the following conditions with general, possibly non-linear, reaction rates r_i :

$$[\mathbf{j}_i \cdot \mathbf{n}]_{\Gamma_f}^{\Gamma_s} = r_i(c_J, c_{J+1}, \dots, c_M) \quad \mathbf{x} \in \Gamma \text{ for } i = J, \dots, M, \tag{116}$$

$$[c_i]_{\Gamma_f}^{\Gamma_s} = 0 \quad \mathbf{x} \in \Gamma \text{ for } i = J, \dots, M. \tag{117}$$

We begin by writing \mathbf{j}_i in full and use $\mathbf{n} = -\mathbf{n}'$ to give $\mathbf{j}_i|_{\Gamma_f} \cdot \mathbf{n} = -\mathbf{j}_i|_{\Gamma_f} \cdot \mathbf{n}'$, a more acceptable form for OpenFOAM®. Since formulating the necessary Robin conditions is the same for all species i , we omit the subscript from here onward unless necessary. Instead we introduce two new subscripts s and f denoting evaluation on the solid or fluid side of Γ :

$$\begin{aligned}
 &-D_i \nabla_{\mathbf{n}}c_s + (\mathbf{u}_f \cdot \mathbf{n}')c_f - D_i \left(\nabla_{\mathbf{n}'}c_f + \frac{z_i F}{RT} \nabla_{\mathbf{n}'}\phi_f \right) \\
 &= r_i(c_J, c_{J+1}, \dots, c_M). \tag{118}
 \end{aligned}$$

We next use the first order approximation $\nabla_{\mathbf{n}'}c_f = B_f(c_f - c_{f,c})$, where the coefficient B_f is determined by OpenFOAM® and $c_{f,c}$ denotes the cell centre value of cells sharing a face with Γ . Alongside the continuity condition Eq. (117) this allows us to rearrange Eq. (118) into a form close to Robin:

$$\begin{aligned}
 &-D_i \nabla_{\mathbf{n}}c_{s,f} = c_{s,f} \left[-(\mathbf{u}_{f,f} \cdot \mathbf{n}') + D_i B_f + \frac{z_i D_i F}{RT_{f,f}} \nabla_{\mathbf{n}'}\phi_{f,f} \right] \\
 &-D_i B_f c_{f,c} + r_i(c_{j=J, \dots, M}). \tag{119}
 \end{aligned}$$

Since r_i may be non-linear, we linearise about an initial guess $\mathbf{c}^N = (c_J^N, c_{J+1}^N, \dots, c_i^N, \dots, c_M^N)$ of concentrations and define $\delta c_j = c_j - c_j^N$:

$$\begin{aligned}
 r_i(c_{j=J, \dots, M}) &= r_i(\mathbf{c}^N) + \delta c_J \left. \frac{\partial r_i}{\partial c_J} \right|_{\mathbf{c}^N} + \delta c_{J+1} \left. \frac{\partial r_i}{\partial c_{J+1}} \right|_{\mathbf{c}^N} + \dots \\
 &+ \delta c_i \left. \frac{\partial r_i}{\partial c_i} \right|_{\mathbf{c}^N} + \dots \\
 &+ \delta c_M \left. \frac{\partial r_i}{\partial c_M} \right|_{\mathbf{c}^N} + \mathcal{O}(\delta c^2). \tag{120}
 \end{aligned}$$

Substitute Eq. (120) into Eq. (119) and take $c_s = c_s^{N+1}$ to be the next iterative solution. To decouple the iterative conditions, all other concentrations we set as $c_j = c_j^N$ for $i \neq j$. The result is the following iterative Robin condition for c_s^{N+1} , the concentration of species i on the solid side of Γ :

$$-D_s^* \nabla_{\mathbf{n}}c_s^{N+1} = -K_s^* c_s^{N+1} - F_s^*, \tag{121}$$

$$D_s^* = D_i, \quad (122)$$

$$K_s^* = \mathbf{u}_f \cdot \mathbf{n}' - D_i B_f - \frac{z_i D_i F}{RT} \nabla_{\mathbf{n}'} \phi_f - \frac{\partial r_i}{\partial c_i} \Big|_{c^N}, \quad (123)$$

$$F_s^* = D_i B_f c_{f,c} - r_i(c^N) + c_s^N \frac{\partial r_i}{\partial c_i} \Big|_{c^N}. \quad (124)$$

The same process can be done to determine the iterative Robin condition for $c_f = c_f^{N+1}$, the concentration of species i on the fluid side of Γ :

$$-D_f^* \nabla_{\mathbf{n}'} c_f^{N+1} = -K_f^* c_f^{N+1} - F_f^*, \quad (125)$$

$$D_f^* = D_i, \quad (126)$$

$$K_f^* = \mathbf{u}_f \cdot \mathbf{n}' - D_i B_s - \frac{z_i D_i F}{RT} \nabla_{\mathbf{n}'} \phi_f - \frac{\partial r_i}{\partial c_i} \Big|_{c^N}, \quad (127)$$

$$F_f^* = D_i B_s c_{s,c} - r_i(c^N) + c_f^N \frac{\partial r_i}{\partial c_i} \Big|_{c^N}. \quad (128)$$

References

- Alizadeh A, Hsu W, Wang M, Daiguji H (2021) Electroosmotic flow: From microfluidics to nanofluidics. *Electrophoresis* 42(7–8):834–868. <https://doi.org/10.1002/elps.202000313>. (ISSN 0173-0835)
- Baird JK (1999) A generalized statement of the law of mass action. *J Chem Educ* 76:1146–1150
- Basu HS, Bahga SS, Kondaraju S (2020) A fully coupled hybrid lattice Boltzmann and finite difference method-based study of transient electrokinetic flows. *Proc Roy Soc A Math Phys Eng Sci* 476(2242):20200423. <https://doi.org/10.1098/rspa.2020.0423>. (ISSN 1364-5021)
- Berg P, Findlay J (2011) Analytical solution of the Poisson–Nernst–Planck–Stokes equations in a cylindrical channel. *Proc Roy Soc A Math Phys Eng Sci* 467(2135):3157–3169. <https://doi.org/10.1098/rspa.2011.0080>. (ISSN 1364-5021)
- Boccardo G, Crevacore E, Passalacqua A, Icardi M (2020) Computational analysis of transport in three-dimensional heterogeneous materials. *Comput Vis Sci* 23(1):1–15
- Chang R (2000) *Physical chemistry for the chemical and biological sciences*. University Science Books, California, p 2000
- Constantin P, Ignatova M (2019) On the Nernst–Planck–Navier–Stokes system. *Arch Ration Mech Anal* 232(3):1379–1428. <https://doi.org/10.1007/s00205-018-01345-6>. (ISSN 0003-9527)
- Dreyer W, Gohlke C, Müller R (2013) Overcoming the shortcomings of the Nernst–Planck model. *Phys Chem Chem Phys* 15(19):7075. <https://doi.org/10.1039/c3cp44390f>. (ISSN 1463-9076)
- Driscoll TA, Hale N, Trefethen LN (2014) *Chebfun Guide*, Pafnuty Publications, Oxford, 2014
- Gagneux G, Millet O (2016) Survey on properties of Nernst–Planck–Poisson system. Application to ionic transport in porous media. *Appl Math Model* 40:846–858
- Icardi M, Barnett R (2021) F Mucicchi spnpFoam. <https://doi.org/10.5281/zenodo.4973896>
- Icardi M, Pescimoro E, Mucicchi F, Hidalgo JH (2022) Computational framework for complex flow and transport in heterogeneous porous media, vol 12. [arXiv:2212.10961](https://arxiv.org/abs/2212.10961) [math.NA]
- Jensen M, Weerd KD, Johannesson B, Geiker M (2015) Use of a multi-species reactive transport model to simulate chloride ingress in mortar exposed to NaCl solution or sea-water. *Comput Mater Sci* 105:75–82
- Ji S, Liu W (2012) Poisson–Nernst–Planck systems for ion flow with density functional theory for hard-sphere potential: I–V relations and critical potentials. Part I: analysis. *J Dyn Differ Equ* 24(4):955–983. <https://doi.org/10.1007/s10884-012-9277-y>. (ISSN 1040-7294)
- Kontturi K, Murtoimäki L, Manzanares JA (2008) *Ionic transport processes*. Oxford Press, Oxford
- Krishna R, Wesselingh J (1997) The Maxwell–Stefan approach to mass transfer. *Chem Eng Sci* 52(6):861–911. [https://doi.org/10.1016/S0009-2509\(96\)00458-7](https://doi.org/10.1016/S0009-2509(96)00458-7). (ISSN 00092509)
- Lai W, Ciucci F (2011) Mathematical modeling of porous battery electrodes—revisit of Newman’s model. *Electrochim Acta* 56:4369–4377
- Latz A, Zausch J (2011) Thermodynamic consistent transport theory of Li-ion batteries. *J Power Sci* 196:3296–3302
- Li H, Toschi F (2020) Plasma-induced catalysis: towards a numerical approach. *Philos Trans Roy Soc A Math Phys Eng Sci* 378(2175):20190396. <https://doi.org/10.1098/rsta.2019.0396>. (ISSN 1364-503X)
- Liu J, Eisenberg B (2014) Poisson–Nernst–Planck–Fermi theory for modeling biological ion channels. *J Chem Phys* 141(22):22D532
- Ma M, Xu Z, Zhang L (2020) Modified Poisson–Nernst–Planck model with Coulomb and hard-sphere correlations. <http://arxiv.org/abs/2002.07489>
- Maex R (2017) On the Nernst–Planck equation. *J Integr Neurosci* 16(1):73–91. <https://doi.org/10.3233/JIN-170008>. (ISSN 1757448X)
- Moshtarihah S, Oppers NA, de Groot MT, Keurentjes JT, Schouten JC, van der Schaaf J (2017) Nernst–Planck modeling of multicomponent ion transport in a Nafion membrane at high current density. *J Appl Electrochem* 47:51–62
- Mucicchi F, Di Pasquale N, Dentz M, Icardi M (2021) Heterogeneous multi-rate mass transfer models in OpenFOAM®. *Comp Phys Commun* 261:107763. <https://doi.org/10.1016/j.cpc.2020.107763>. (ISSN 00104655)
- Němeček J, Kruijs J, Koudelka T, Krejčí T (2018) Simulation of chloride migration in reinforced concrete. *Appl Math Comput* 319:575–585
- Nernst W (1888) Zur kinetik der in lösung befindlichen körper. *Phys Chem* 2:613–637
- Nuca R, Stovrik E, Radu FA, Icardi M (2022) Splitting schemes for coupled differential equations: Block schur-based approaches and partial Jacobi approximation, vol 12. <https://arxiv.org/pdf/2212.11111.pdf>
- Patankar SV, Spalding DB (1972) A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int J Heat Mass Transf* 15(10):1787–1806. [https://doi.org/10.1016/0017-9310\(72\)90054-3](https://doi.org/10.1016/0017-9310(72)90054-3). ISSN 00179310
- Pimenta F, Alves M (2019) A coupled finite-volume solver for numerical simulation of electrically-driven flows. *Comput Fluids* 193:104279. <https://doi.org/10.1016/j.compfluid.2019.104279>. (ISSN 00457930)
- Planck M (1890) Ueber die Erregung von Electricität und Wärme in Electrolyten. *Ann Phys Chem* 275(2):161–186. <https://doi.org/10.1002/andp.18902750202> (ISSN 00033804)
- Probstein RF (1994) *Physicochemical hydrodynamics*, 2nd edn. Wiley-Interscience, New Jersey (ISBN 0-471-01011-1)
- Psaltis STP, Farrell TW (2011) Comparing charge transport predictions for a ternary electrolyte using the Maxwell–Stefan and Nernst–Planck equations. *J Electrochem Soc* 158:A33
- Richardson G, Foster J, Ranom R, Please C, Ramos A (2020) Charge transport modelling of lithium ion batteries. [arXiv:2002.00806](https://arxiv.org/abs/2002.00806)

34. Samson E, Marchand J (1999) Numerical solution of the extended Nernst–Planck model. *J Colloid Interface Sci* 215(1):1–8. ISSN 00219797. <https://doi.org/10.1006/jcis.1999.6145>
35. Schmuck M (2011) Modeling and deriving porous media Stokes–Poisson–Nernst–Planck equations by a multi-scale approach. *Commun Math Sci* 9:685–710
36. Squires TM (2009) Induced-charge electrokinetics: fundamental challenges and opportunities. *Lab on a Chip* 9(17):2477. <https://doi.org/10.1039/b906909g> (ISSN 1473-0197)
37. Voukadinova A, Gillespie D(2019) Energetics of counterion adsorption in the electrical double layer. *J Chem Phys* 150(15):154706. <https://doi.org/10.1063/1.5087835> (ISSN 0021-9606)
38. Yang P, Sant G, Neithalath N (2017) A refined, self-consistent Poisson–Nernst–Planck (PNP) model for electrically induced transport of multiple ionic species through concrete. *Cement Concr Compos* 82:80-94
39. Ying J, Fan R, Li J, Lu B (2021) A new block preconditioner and improved finite element solver of Poisson–Nernst–Planck equation. *J Comput Phys* 430:110098. <https://doi.org/10.1016/j.jcp.2020.110098> (ISSN 00219991)
40. Zheng Q, Wei G (2011) Poisson–Boltzmann–Nernst–Planck model. *J Chem Phys* 134
41. Zheng Q, Wei G (2011) Poisson–Boltzmann–Nernst–Planck model. *J Chem Phys* 134:194101
42. Zhu H, Kee R (2016) Membrane polarization in mixed-conducting ceramic fuel cells and electrolyzers. *Int J Hydrogen Energy* 41:2931–2943

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.