

Components for atomistic-to-continuum multiscale modeling of flow in micro- and nanofluidic systems

Helgi Adalsteinsson^{a,*}, Bert J. Deusschere^a, Kevin R. Long^b and Habib N. Najm^a

^a Sandia National Laboratories, Livermore, CA 94551, USA

^b Department of Mathematic and Statistics, Texas Tech University, Lubbock, TX 79409, USA

Abstract. Micro- and nanofluidics pose a series of significant challenges for science-based modeling. Key among those are the wide separation of length- and timescales between interface phenomena and bulk flow and the spatially heterogeneous solution properties near solid-liquid interfaces. It is not uncommon for characteristic scales in these systems to span nine orders of magnitude from the atomic motions in particle dynamics up to evolution of mass transport at the macroscale level, making explicit particle models intractable for all but the simplest systems.

Recently, atomistic-to-continuum (A2C) multiscale simulations have gained a lot of interest as an approach to rigorously handle particle-level dynamics while also tracking evolution of large-scale macroscale behavior. While these methods are clearly not applicable to all classes of simulations, they are finding traction in systems in which tight-binding, and physically important, dynamics at system interfaces have complex effects on the slower-evolving large-scale evolution of the surrounding medium. These conditions allow decomposition of the simulation into discrete domains, either spatially or temporally. In this paper, we describe how features of domain decomposed simulation systems can be harnessed to yield flexible and efficient software for multiscale simulations of electric field-driven micro- and nanofluidics.

Keywords: Multiscale, multiphysics, electrodynamic transport, microfluidics, desalination, puppeteer

1. Introduction

The term multiscale simulations is commonly used to describe model systems whose characteristic scales (typically spatial or temporal) permit the system simulation to be divided into two or more separate simulation models linked through a handshake bridge. Multiscale simulations are finding increasing use for modeling systems where the actions of small-scale, tight-binding events have looser coupling to larger-scale system evolution. Multiscale multiphysics simulations (sometimes referred to as “hybrid methods”) apply disparate physics models for fine-grained and coarse-grained regions of the multiscale simulation. Since these types of simulations are the focus of this paper, we will from here on use the term “multiscale” to mean “multiscale multiphysics”.

Multiscale simulations can be coarsely divided into two main categories. In concerted multiscale simulations, the differently represented sections of the sim-

ulation system are evolved together, with information transfer performed between the models at every step. In stepwise multiscale simulations, the models progress independently, and the simulation results from each stage are used as initial parameters or boundary conditions for subsequent stages. Each of these approaches is appropriate for different simulation types; concerted multiscale simulations permit very tight integration of coarse-grained models into fine-grained simulations while stepwise multiscale simulations are better suited for providing fine-grained foundations for long-timescale coarse-grained simulations.

Multiscale simulations have gained significant traction in select simulation fields such as coupled quantum-mechanical/molecular mechanics (QM/MM) systems [1]. Domain decomposition for multiscale simulations of fluidic systems was explored by Hadjiconstantinou [2], who showed that high efficiencies can be achieved when a multiscale system is appropriate for the Schwarz alternating method [3], in which the problem is divided into subdomains that are solved

*Corresponding author. E-mail: hadalst@sandia.gov.

in an alternating fashion until convergence is reached. Later research applied similar methods to various microfluidic and nanofluidic simulation systems [4,5]. For a thorough review of multiscale phenomena in micro- and nanofluidics, the reader is directed to a recent review by Hu and Li [6].

This paper describes a design strategy for composing multiscale simulations out of existing single-physics simulation components. The goal is to make use of current, fully validated simulation codes for multiscale simulations and to restrict modifications to these packages such that their numerical behavior and performance are not adversely affected. Two examples are shown using Schwarz decomposition for multiscale simulations involving solid-liquid boundaries, demonstrating how the multiscale coupling between the systems can be handled in a way that is minimally invasive on the domain solutions using a combination of system setup, simulation boundary conditions, and simple callbacks managed by a “mediator” component to handle data translation and convergence management. It is further shown how this design matches well with the design decisions made in the Sundance [7] and PST [8] component toolkits, allowing construction of multiscale multiphysics simulations software without requiring significant changes to existing simulation components.

2. Multiscale approach

Consider a multiscale simulation system consisting of a large continuum simulation with one or more nested atomistic simulation regions. We can now classify physics modeled in the system into three categories:

- 1.1 Interactions completely contained within a single simulation region.
- 1.2 Interactions crossing simulation regions that can be confined to a handshake region shared by two simulation regions (e.g. boundary concentrations and mass transport).
- 1.3 Interactions that cross simulation regions but cannot be confined to a handshake region (e.g. electric fields).

The systems considered in this paper pose a set of challenges for the multiscale simulation:

- 2.1 The system timescale is orders of magnitude greater than the durations of individual atomistic simulations. Time integration, thus, re-

quires non-uniform progression of simulation time.

- 2.2 Each nested atomistic simulation is orders of magnitude more expensive than the continuum evaluation. To complete the simulation in a reasonable amount of time, atomistic results need to be reused or interpolated whenever possible.
- 2.3 If the atomistic simulations have to share electric field information with the continuum domain, there may be limits to the total perturbation the atomistic system is permitted to undergo between updates to the continuum system.
- 2.4 For efficient sampling, atomistic simulations commonly need to be run in replicate. This poses constraints on when and how the continuum model can be updated with particle information.

It is easy to envision a high-level algorithm for running these multiscale simulations (Algorithm 1).

The challenge, however, is to implement the algorithm in an efficient manner on parallel platforms with minimal requirements for reinventing or rewriting existing simulation software. This challenge is addressed here in a component-driven manner as shown in the control flow diagram in Fig. 1.

This paper will demonstrate how the introduction of a select few design constraints and a high-level “puppeteer” component that handles interfacing and data translation can greatly facilitate the design and implementation of simulation software from existing single-physics simulation software.

The design constraints imposed on the system can be enumerated as follows:

```

for each outer (controller) iteration do
  Identify nested simulations
  if simulation can be interpolated then
    yield interpolated value
  else
    for each nested simulation do
      Run nested simulation
      Store value in interpolation table
    yield simulation results
  end for
end if
  Gather and process nested simulation results
  Update controller simulation
end for

```

Algorithm 1. Pseudocode overview of a multiscale simulation.

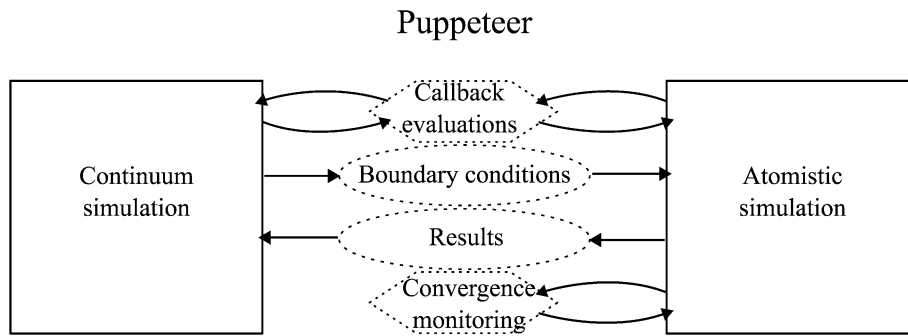


Figure 1. Algorithm 1 represented with a component diagram. Labels inside dotted regions identify functionality managed by the puppeteer component.

- 3.1 The puppeteer simulation component has a view of all simulation input data, takes care of transfer of information between components, translation between data layouts, scales, or representations, and testing for termination criteria such as convergence or error conditions.
- 3.2 The low-level simulation components interact only with the puppeteer and not directly with “peers” within the simulation.
- 3.3 Overall time progression is handled by the puppeteer; low-level simulations get notified of initial simulation time (if needed) at the start of each iteration.
- 3.4 When a low-level simulation component provides callbacks to extract shared information, the order in which these callbacks is made must not affect the results of the simulations.

Although it should, in principle, be easy to have recursively nested puppeteer simulation component, the designs explored in this study had a single puppeteer communicating with continuum and atomistic simulations. The puppeteer simulation components were written in Python, while the continuum and atomistic simulation components used C++, C and Fortran77. Bindings between the puppeteer and the lower-level components were generated using SWIG [9], using `directors`¹ to permit cross-language polymorphism for callback interfaces.

This software design enabled us to set up the simulation system using a Schwarz decomposition, with the added wrinkle that the low-level simulation components may, through callbacks, require intermediate evaluations from the controller simulation. Further, this design aided in the efficient utilization of parallel computer platforms by easing the task of subdividing

a system of distributed-memory nodes into subgroups for individually parallel replicate simulations (Fig. 2).

In the following sections, this general design is applied to two important model systems involving electrophoretic transport. Section 3.2, uses simulations of electrostatics in microfluidic channels to highlight features of multiscale models with very-large separation of scales, while Section 3.3 addresses a system with relatively significant overlap of scales to highlight handling of callbacks for field evaluations and convergence testing.

3. Application to electrodynamic transport

3.1. Multiscale challenges

This work addresses the use of multiscale multiphysics simulations for two types of electrodynamic flow; microfluidics and electrodesalination. Both systems have substantial features in common, making them suitable candidates for the development of multiscale simulation components.

- Tight-binding atomistic regions are located at solid-liquid interfaces, providing guidance for spatial decomposition of the simulation systems.
- Driving force for the systems comes from applied electric fields originating at discrete locations and affecting all locations of the simulation systems.
- The desirable macroscale behaviors in the system arises from the interactions of charged particles in electric bilayers at the solid-liquid interfaces with the applied electric fields.
- Related systems have been well studied using computational fluid dynamics methods.

¹Added in SWIG-1.3.20.

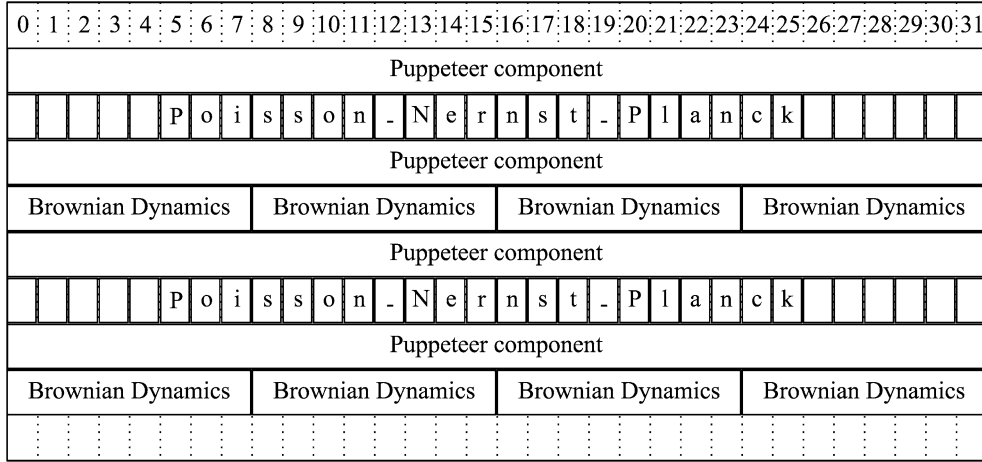


Figure 2. Schematic management of a multiscale simulation on 32 nodes with MPI-unaware continuum simulations and four replicate BD simulations (from Section 3.3).

- Both continuum and atomistic methods have been applied to a range of related systems, including small-scale models of microfluidic systems and biological ion channels.

At the heart of these systems is the electric double layer formed in a polarizable medium exposed to an electrostatic field. The characteristic width of this double layer is commonly described using the Debye length as a measure of the decay length for screening out of electric fields by mobile charges. In a solution containing mobile ions, the Debye length κ^{-1} is:

$$\kappa^{-1} = \sqrt{\frac{\epsilon\epsilon_0 k_B T}{2N_A e^2 I}}, \quad (1)$$

where ϵ is the dielectric constant, ϵ_0 is the electric permittivity in vacuum, k_B is Boltzmann's constant, T is the solution temperature, N_A is Avogadro's number, e is the electron charge and I is the ionic strength of the electrolyte given by

$$I = \frac{1}{2} \sum_{i=1}^A c_i q_i \quad (2)$$

in which A is the number of ionic species in solution, c_i is the solution concentration of species i and q_i is its charge number.

As the electrolytes in this double layer become more concentrated and less like the bulk aqueous environment, e.g. in tightly confined spaces, high charge on the solid surface, or high salt concentrations, core approximations behind the continuum model become less

appropriate (for a debate on when and how this breakdown occurs, see [10–12]). To address this, successor models must account for finite particle sizes, non-uniform viscosity of water under these conditions, and increasingly ordered structure arising in both water and electrolyte [13]. While some of these can be simulated using Lattice Boltzmann methods [14] or classical density functional theory [15], atomistic methods are still the most common way of addressing the increased system complexity using system-agnostic parameters [16,17].

The following sections illustrate how atomistic modeling can be coupled with continuum model to selectively model the physics at charged solid-liquid interfaces while giving insight into behavior and evolution at system-scale.

3.2. Electroosmotic flow

Electroosmotic flow is the transport of a solution (generally water) containing an excess charge (matched by a countercharge in its surrounding solid environment) is subjected to an electric potential. As ions in the solution migrate due to the applied field, the bulk solution gets pulled along due to viscous drag. Electroosmotic flow is a foundation for microfluidics, which have become a cornerstone in systems to efficiently manipulate small amounts of fluids, such as chemical analysis, detection systems, inkjet printers, and biomedical assay systems. Microfluidic systems have been the subject of extensive modeling using continuum methods [5,18,19] and particle simulations [20].

Electrokinetic microfluidic systems are conceptually quite simple. One or more narrow channels ($\ll 1$ mm) are patterned into a solid material (e.g. glass, silicon, or polycarbonate) and connected to at least one fluid-filled reservoir. As the fluid fills the channel, proton exchange between the fluid and chemical groups on the channel walls results in a charge separation between the fluid and solid, which resides in an ionic double layer in the solution closest to the wall. When an electric potential is applied along the fluid-filled channel, the net charge of this double layer gets “dragged” along the electric field (electroosmotic flow), and the rest of the solution in the channel gets pulled along due to solvent viscosity. This implies that in order to predict flow characteristics in microfluidic systems, a model system must tie together atomistic-level insight into the formation and behavior of the electric double layer near the channel wall and the long-timescale behavior dictated by the geometry and design of the microfluidic system [21,22].

The Helmholtz–Smoluchowski relationship [23] relates the slip velocity \mathbf{u}_{slip} for bulk electroosmosis to the electric potential \mathbf{E} at the slip-plane between the double layer at the interface and the shielded electrostatic potential ζ at the solution slip plane near the wall.

$$\mathbf{u}_{slip} = -\frac{\epsilon\zeta}{\mu}\mathbf{E} \quad (3)$$

so that if the ζ -potential is uniformly constant, the electroosmotic velocity will be proportional to the electric field everywhere [24]. In general, though, microchannels may well have non-uniform charge characteristics, whether non-deliberate or as a design feature used for micromixers [25] or superfast electrophoresis [26].

The detailed characteristics of the electric double layer arise from an interplay between wall charges and the ionic solution [21]. The Gouy–Chapman model (expressed through the Grahame equation) does a reasonable job predicting electrostatic potential at a charged membrane in solution when the charge on the membrane is sufficiently low:

$$\sigma^2 = 2\epsilon\epsilon_0RT \sum_{i=1}^N \left(C_{i\infty} e^{\frac{-z_i E_0 F}{RT}} - 1 \right), \quad (4)$$

where σ is the charge density of the membrane surface, ϵ is the dielectric constant of the medium, ϵ_0 is the dielectric permittivity in vacuum, N is the number

of ionic species in solution, $C_{i\infty}$ is the concentration of ionic species i at infinite distance from the wall, E_0 is the electrical potential at the membrane surface, F is Faraday’s constant, R is the gas constant, and T is the temperature. The Stern modification to the Gouy–Chapman model takes the finite sizes of solute ions into account, creating a region containing no countercharges (the Stern layer) adjacent to the charged surface. While the Grahame equation and the Gouy–Chapman–Stern formulation can give reasonable estimates of electrostatic potential in an electrolyte solution near a charged membrane, although it cannot account for the effects of localized variances in wall charge or chemical composition as first-principles parameters for continuum simulations of electroosmotic flow in microchannels with non-uniform wall characteristics [21], the inverse has traditionally not been true, and it is only very recently that quantitative measures of membrane charge in solution became possible [27].

3.2.1. Formulation

Continuum model: If the microchannel is sufficiently wide (100 nm–100 μ m) [6], electroosmotic flow in microfluidics can be described using the Navier–Stokes equations for incompressible fluids [28]:

$$\nabla \cdot \mathbf{u} = 0, \quad (5)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u}, \quad (6)$$

where \mathbf{u} is the flow velocity vector, p is the pressure normalized by density and ν is the kinematic viscosity.

We have previously investigated continuum simulations of microfluidics simulations with chemically reacting species [29]. In this study, we neglect chemical reactions and focus on water containing non-reactive electrolytes. In this case, the transport of solute species is governed by [28]:

$$\frac{\partial c_i}{\partial t} + \nabla \cdot [c_i(\mathbf{u} + \mathbf{u}_i^e)] = \nabla \cdot (D_i \nabla c_i), \quad (7)$$

where c_i is the concentration of species i and D_i is the corresponding diffusivity. The electromigration velocity \mathbf{u}_i^e accounts for the electrophoretic movement of electrically charged species relative to the bulk flow. This velocity is given by [28]

$$\mathbf{u}_i^e = -\beta_i z_i F \nabla \phi \quad (8)$$

where β_i is the electrophoretic mobility for species i , z_i is the charge number, F is the Faraday constant

(9.648×10^4 C/mol) and ϕ is the electrostatic field potential.

Allowing for concentration field gradients, the electrostatic field potential, ϕ can be obtained from the current continuity constraint [28],

$$\nabla \cdot (\sigma \nabla \phi) = -F \sum_i z_i \nabla \cdot (D_i \nabla c_i) \quad (9)$$

as addressed in [29]. For simplicity, this study uses a constant electrostatic field throughout the simulation system, but the multiscale setup would be identical for a simulation with spatially dependent field strengths.

The diffusivity D_i and the mobility β_i are coupled through the Nernst–Einstein equation [28]

$$D_i = RT\beta_i, \quad (10)$$

where R is the universal gas constant and T the temperature.

Since the double layer is delegated to the atomistic model, the continuum system is assumed to satisfy the electroneutrality condition

$$\sum_i z_i c_i = 0 \quad (11)$$

everywhere in its domain.

Atomistic simulation: The molecular dynamics simulations solve Newton's equations of motion for N interacting particles:

$$m_i \frac{\partial^2 \mathbf{r}_i}{\partial t^2} = \mathbf{F}_{i,t}, \quad i = 1, \dots, N, \quad (12)$$

where $\mathbf{F}_{i,t}$ is the force acting on particle i with coordinate \mathbf{r}_i and mass m_i at time t . For this study we chose the Gromos96 force field, which has been fully documented elsewhere [30]. Time integration of the MD system was performed using the Leapfrog algorithm and heat gained through viscous friction was removed using a Berendsen thermostat [31]:

$$\frac{\partial \mathbf{v}}{\partial t} = \frac{\mathbf{F}}{m} + \frac{1}{2\tau} \left(\frac{T}{T_0} - 1 \right) \mathbf{v}, \quad (13)$$

where T_0 is the target temperature (300 K) and τ is the time coupling constant for the velocity rescaling (set to 0.1 ps).

As mentioned previously, the entire effect of the electric double layer is handled by the MD simulations,

which also has to satisfy the electroneutrality condition (11). Thus, the net charge on the segment of the channel wall in the MD simulation system is countered by an equal and opposite solution charge from the electrolyte. To ensure that the atomistic simulation system was sufficiently large to contain the double layer, MD simulations were performed on model systems of varying size and the population densities of electrolyte ions relative to the wall were computed using the radial distribution function

$$g_{AB}(r) = \frac{\langle \rho_B(r) \rangle}{\langle \rho_B \rangle_{local}} \quad (14)$$

$$= \frac{1}{\langle \rho_B \rangle_{local}} \frac{1}{N_A} \times \sum_{i \in A} \sum_{j \in B} \frac{\delta(r_{ij} - r)}{4\pi r^2}, \quad (15)$$

where $\langle \rho_B(r) \rangle$ is the particle density of particles in B at distance r from particles in A and $\langle \rho_B \rangle_{local}$ is a normalization factor (the average density of particles in B around particles in A).

These radial distribution functions provide a measure of the net charge distribution in the simulation system. To force the net overcharge to zero at a specific distance d , the simulation system is made symmetric across a plane parallel to the wall section. In practice, this means that each MD simulation contains two walls, spaced $2d$ apart, as illustrated in Fig. 3.

Each MD simulation was constructed as a slab region, periodic along x and y . The z -edges of the simulation box were capped with a silicon lattices cut along the (001) vector. Unfilled valences facing toward the interior of the channel were capped with hydroxyl groups and unfilled valences facing away from the channel were capped with hydrogen atoms. Partial charges were assigned to the first few layers of atoms based on *ab initio* Hartree–Fock electronic structure calculations with the 6-31G(d,p) basis set and natural population analysis [32] performed using the Massively Parallel Quantum Chemistry Program MPQC [33]. Atomic partial charges and bonded force field parameters for the walls are given in Tables 1 and 2 respectively.

For systems with charged walls, excess negative charge was assigned equally to all hydroxyl oxygens along the walls. This charge assignment strategy was previously used by Qiao and Aluru [20] for modeling electroosmotic flow. During dynamics simulations, the

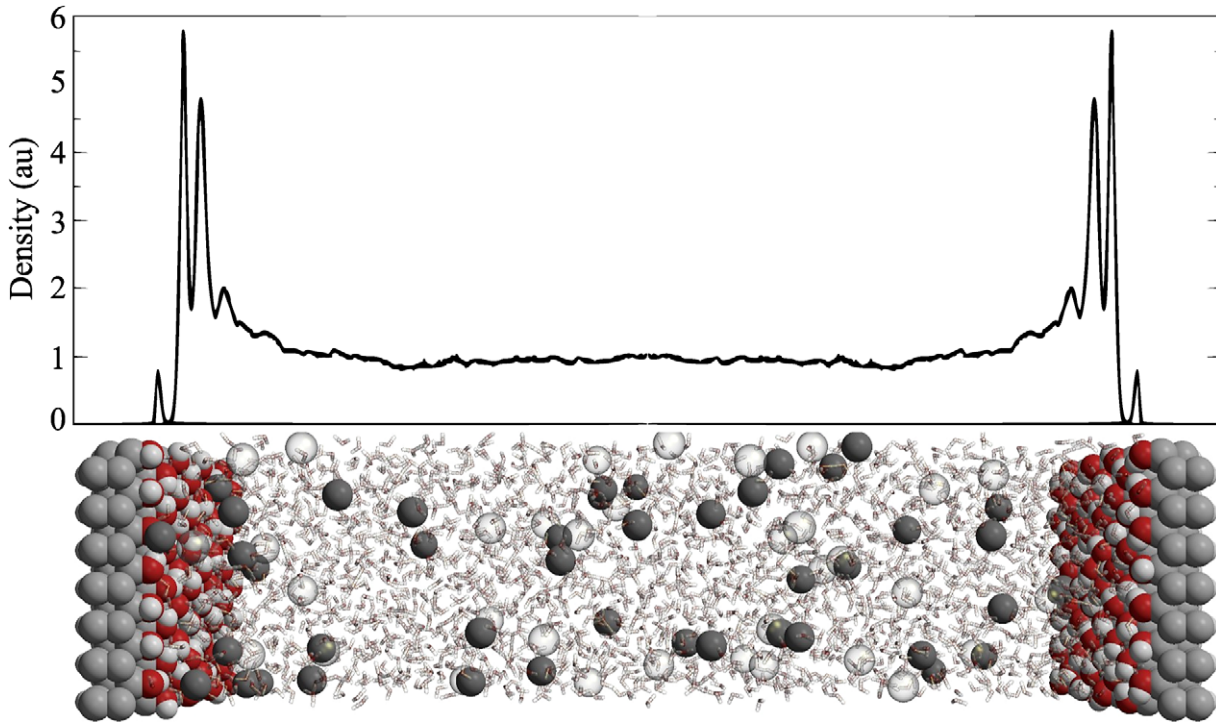


Figure 3. The radial distribution function for potassium in a 10 nm wide model channel with a charge of -0.1 C/m^2 on the walls.

Table 1

Atom types and partial atomic charges for Silicon walls. Atom labels are defined in Fig. 4

Label	Gromacs type	Charge
H(1)	HO	0.4312
O(2)	OA	-0.7340
Si(3)	SI	0.5664
Si(4)	SI	-0.2636
Si(5)	SI	0.0000
Si(6)	SI	0.0000
H(7)	H	0.0000

silane hydrogen atoms were held fixed in space but all other wall atoms were allowed to move. The interior of the slab was filled with SPC/E waters [34], Na^+ and Cl^- ions were added to obtain the desired salt concentration, and excess Na^+ or Cl^- ions were added to counter the net charge on the walls.

Each MD evaluation was run with five to ten replicate simulations for improved sampling. An electric potential was applied along the x-axis and flow rates were calculated by counting fluxes across planes parallel to each of the principal coordinate axes. Each MD simulation was allowed to equilibrate for 30 ps before the external electric field was turned on, followed by a

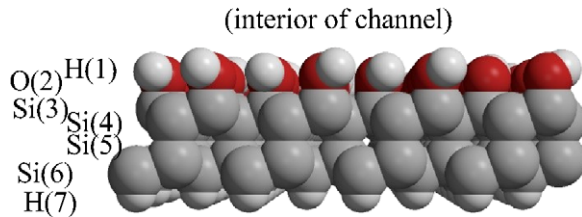


Figure 4. Labels used for the silicon walls.

Table 2

Bonded parameters used for Silicon walls. Atom type assignments are given in Table 1

Bond	b0	kb
Si-Si	0.23500	251040.
Si-H	0.10900	374468.
Si-OA	0.16300	251040.
Angle	th0	cth
Si-Si-Si	109.500	397.480
Si-Si-H	109.500	397.480
Si-Si-OA	109.500	397.480
Si-OA-HO	109.500	397.480
Dihedral	q0	cq
Si-Si	0.000	167.36
Si-OA	0.000	3.766

1 ns equilibration run and a 4 ns production run with an external field.

The simulation system was extended by 2 nm along the z-axis and long-range electrostatics were evaluated using Particle-Mesh Ewald summation [35] with a 1 nm explicit cutoff, fourth-order grid interpolation and 0.12 nm grid spacing.

3.2.2. Multiscale implementation

The multiscale simulation setup is depicted schematically in Fig. 5. The overall simulation is driven by the continuum model, which at each timestep identifies discrete locations along the channel wall at which to request electroosmotic flow rates. The simulation framework allows for table lookups or interpolation of these values, although the interpolation component was not implemented for this study. When the wall conditions differ sufficiently from earlier evaluations, a model builder component gets called to construct an MD system corresponding to the solution conditions requested by the continuum model, MD simulations get run for the system (as described above), and a post-processing component computes flow rates, averages them over the replica simulations and reports the net velocities back to the wall handler.

The implementation of this multiscale scheme requires very little invasion into the individual simulation components:

- The individual MD simulations can be carried out using an unmodified version of the Gromacs simulation program.
- The component for interpolation and tabulation of MD results consisted of several subcomponents to generate input files for the MD simulations, gather simulation results, etc. Most of these required only a few hundred lines of python scripting code and assistance from a few general-purpose particle modeling components.

- The continuum simulation code was unchanged from earlier all-continuum work, except for the introduction of logic to identify and request MD simulations at specific points. All told, these amounted to only a small fraction of the overall simulation program.

If the characteristic timescales for the particle and continuum methods were of a compatible magnitude, the multiscale simulations of these systems could be directly implemented according to the schematic shown in Fig. 6. While this type of multiscale decomposition has been used successfully for dense fluid flows [2] and materials simulations [36], the characteristic timescales for our simulations differ by far too much to permit this direct coupling. To permit coupling between the systems, the simulation takes advantage of the relatively smooth variation of chemical properties and slow time variance of the continuum solution conditions (in time-dependent problems) to perform interpolation both through space (i.e. along the channel wall) and in time.

To perform the interpolation in space, and to pick the locations in space where MD evaluations are requested, the implementation relies on spline interpolations. A key assumption to accomplish this is that smoothly varying input conditions for the MD model will result in smoothly varying MD predictions (an assumption that should hold for most physical systems, unless there are bifurcations in the underlying physics). For situations where this holds, the locations to perform MD evaluations at can be selected as the spline point locations of a spline interpolant that represents the MD input conditions well. Starting with the end points of the interval on which an interpolant is needed, a next spline point is successively chosen as the point where the spline interpolant of the MD input condition profile, based on the previously selected spline points, deviates the most from the true input profile. Once the

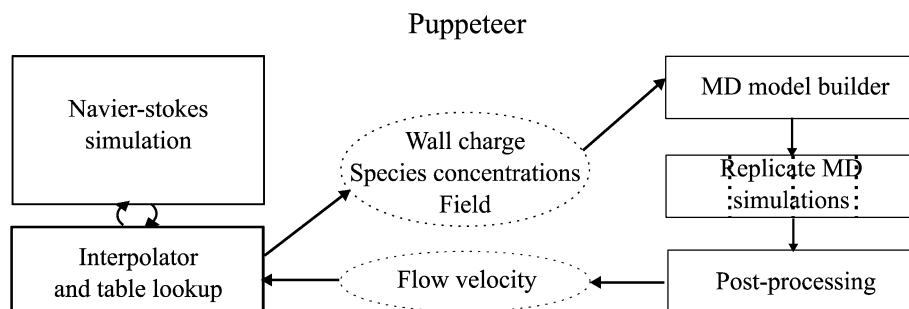


Figure 5. Flow chart for multiscale microfluidic flow simulations.

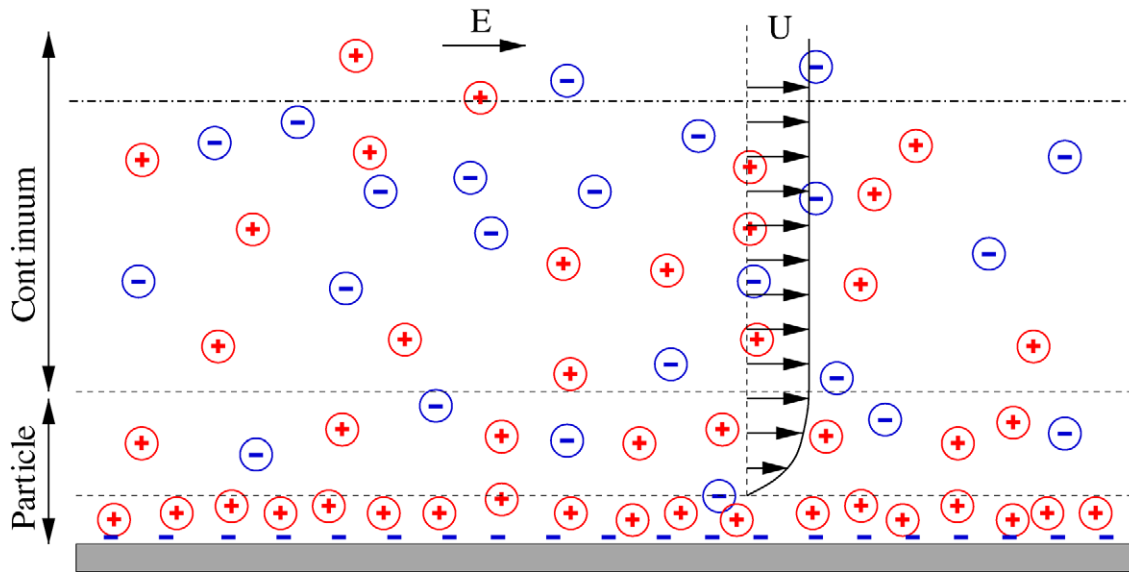


Figure 6. A schematic overview of the division between particle- and continuum models in electrokinetic microfluidics.

spline interpolant approximates the true input condition profiles to within a desired tolerance, MD evaluations are requested, conceptually with the input conditions corresponding to the spline points.

In practice, this scenario needs to be amended slightly, however, to account for the fact that the MD system can only accommodate a finite number of discrete input states, as it has a finite, countable number of particles. For example, the need to balance any wall charge with a discrete, integer number of ions in solution means that only a limited number of charge densities can be realized, with a granularity that depends on the number of particles in the MD system. To account for this, the spline points are restricted to conditions that can be realized with the MD system, and the tolerance on the accuracy of the spline interpolant needs to be consistent with the granularity of the MD systems.

Further, given the high cost of the MD simulations, all MD results are stored in a table, which can be queried for matches between current input conditions and those of previously evaluated systems. This way, only newly encountered situations need to be evaluated, resulting in significant time savings. The extension of this scheme to a full-blown tabulation procedure with interpolation between input conditions is the subject of future work.

The final set of new input conditions is forwarded on to an MD model builder, where simulation systems similar to the one shown in Fig. 3 are constructed to match the desired conditions in the solution and on the wall. Each MD evaluation is performed as a set

of replicate simulations that are run in parallel, post-processed to extract flow rate information, and finally gathered and reported back to the interpolator component.

Upon return from all MD solvers, the desired macro-scale observable (in this case the electroosmotic wall velocity) is reconstructed over the full interval using a spline interpolant through the earlier selected spline points.

3.2.3. Results

For a proof of concept, multiscale simulations were performed of the 2D steady-state flow-field through a straight microchannel, 1 cm long in the streamwise, x -direction and 100 μm deep in the y -dimension. The wall charge along both channel walls was set to a uniform -0.1 C/m^2 with a local increase in magnitude to -0.14 C/m^2 near the middle of the channel as shown in Fig. 7. For simplicity, the electric field strength was assumed to be a constant 0.35 V/nm in the streamwise direction.

In order to perform the multiscale coupling, the wall charge profile was approximated with a cubic spline profile. Using the procedure outlined in Section 3.2.2, this charge profile was approximated to within the resolution of the MD solver with spline points at just 4 unique charge values. For each of these 4 charge values, the corresponding electroosmotic velocity was computed on the microscale with the MD solver component. The wall velocity boundary condition on the continuum level was then recon-

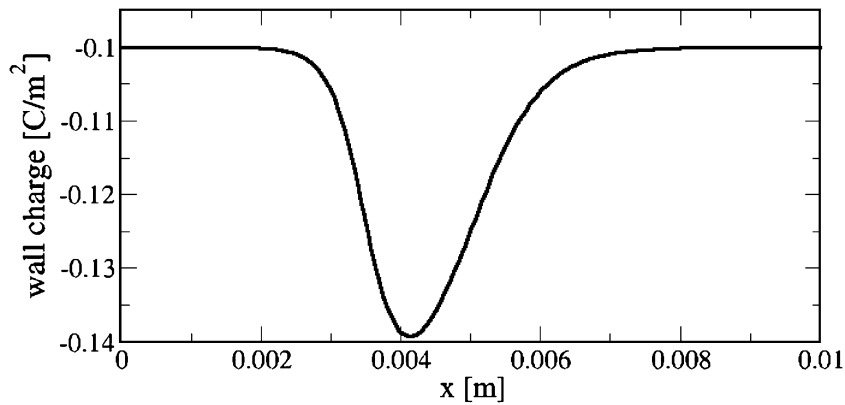


Figure 7. The wall charge density as a function of position along the channel wall.

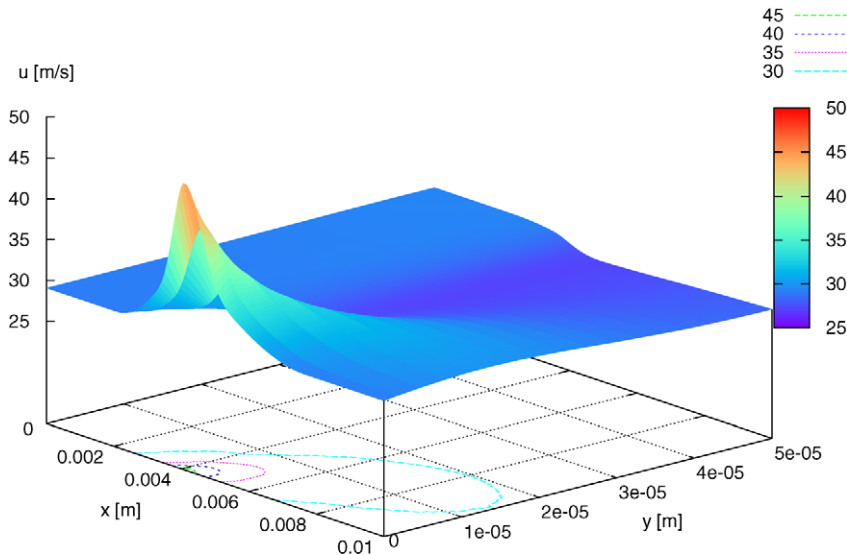


Figure 8. Streamwise velocity profile in bottom half of the domain resulting from the wall charge densities in Fig. 7.

structed with a cubic spline through those velocity data.

Figure 8 shows the resulting 2D streamwise velocity field. Note that since the flow field is symmetric along the centerline, only the bottom half of the domain is shown. Upon inflow to the domain at $x = 0$, the velocity field is uniform. Further downstream, however, the increase in the wall charge magnitude accelerates the electroosmotic flow at the boundary. For continuity reasons, this generates a secondary recirculating flow field, superimposed on the uniform flow field, resulting in a lower velocity near the center of the channel. After some distance, viscosity smoothens the velocity field again, resulting in a near-uniform flow field by the time the channel exit is reached.

The effects of local changes in the electroosmotic flow velocity are important as the secondary flow fields caused by them increase mixing in the microfluidic channel. In some cases, increased mixing is a desired property, and channel designers deliberately adjust wall properties in order to create these recirculation zones. In other cases, increased mixing is to be avoided as it dramatically reduces the separation efficiency of the channel. For either situation, the multi-scale coupling scheme developed here allows prediction of the outcome of variations in wall properties, which is essential for design purposes.

3.3. Nanoporous electro dialysis membranes

Water purification is a large and growing issue across the world. Shortages of water suitable for hu-

man consumption affect a significant percentage of the world's population [37], contributing to rising energy costs [38,39], global instability, and increased rates of death and disease [37].

Water purification reverse osmosis or thermal means are mature technologies with proven scalabilities. However, they require large amounts of energy and may, in many cases, be needlessly exhaustive in the removal of ionic species.

Electrodialysis is the third most common method for large-scale desalination of water [37]. The method relies on electrophoretic transport of solute ions from a diluate (source) stream through ion exchange membranes and into a concentrate (sink) stream. This approach, thus, differs from reverse osmosis and thermal desalination methods in that solute ions are removed from a water stream, rather than the other way around, and that only ionic solutes are removed from the source stream. These characteristics facilitate the design of flow-through electrodialysis systems (shown schematically in Fig. 9), can help reduce the effects of fouling and clogging of desalination membranes, but have also largely limited the methods to water sources with low salinity.

Selective electrodialysis is a relatively recent approach to treatment of marginally impaired water. In selective electrodialysis, the ion exchange membranes are chosen such that they facilitate transport of specific target ions more than those of common, non-toxic ionic solutes. Since the amount of salt transported across the ion exchange membrane is now only a fraction of the overall salt concentration in the source stream, these

systems are expected to achieve substantially higher energy efficiency than reducing the concentrations of all ionic solutes to levels dictated by the least desirable species [40].

A core requirement for the design of selective electrodialysis systems is availability or design of selective ion exchange membranes. In this section, we address simulation methods to study selective ion transport through nanoporous membranes, using atomistic simulation methods to account for the interactions of solute ions with the membrane material and continuum methods to address the effects of applied potential, concentration polarization, and other system-scale effects.

3.3.1. Formulation

Continuum model: The Poisson–Nernst–Planck (PNP) model [41] has become a key simulation method for ion channels. It consists of a transport equation for the species concentrations c_i :

$$\frac{\partial c_i}{\partial t} + \nabla \cdot [c_i(\mathbf{u} + \mathbf{u}_i^e)] = \nabla \cdot (D_i \nabla c_i) \quad (16)$$

coupled with the Maxwell equation for the electric potential ϕ :

$$\nabla \cdot (\epsilon \nabla \phi) = -\rho_e - \rho_p, \quad (17)$$

where D_i is the diffusion coefficient, \mathbf{u} is the bulk convective velocity field, ρ_e is the ionic charge density in the solution, computed as $\rho_e = F \sum_i z_i c_i$, ρ_p is the density of fixed charges (e.g., on the membrane), ϵ is

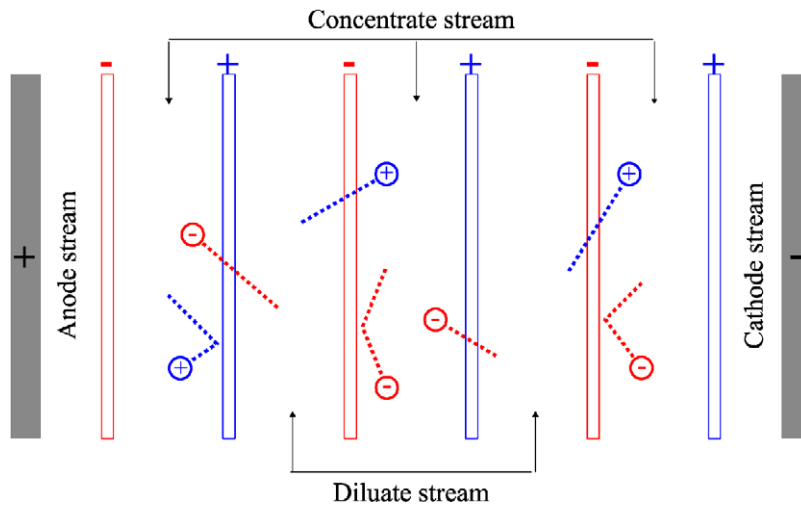


Figure 9. A schematic of an electrodialysis stack. Negatively charged (cation exchange) membranes are open to positive ions, positively charged (anion exchange) membranes are open to negative ions.

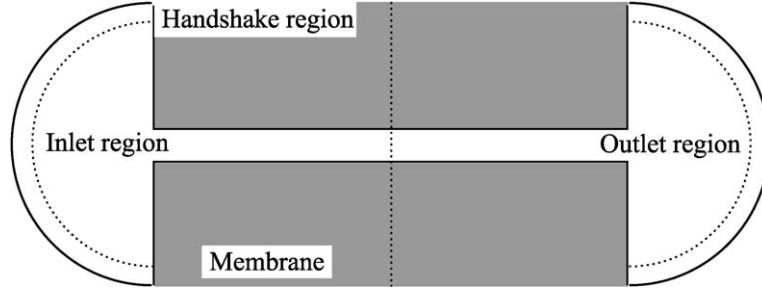


Figure 10. Schematic drawing of the BD simulation geometry for a nanopore.

the electrical permittivity of the medium and \mathbf{u}_i^e is the electrophoretic velocity of species i , obtained from

$$\mathbf{u}_i^e = -\beta_i z_i F \nabla \phi, \quad (18)$$

where z_i is the valence, F is the Faraday constant, and β_i is the electrophoretic mobility, related to the diffusion coefficient through the Nernst–Einstein equation: $D_i = RT\beta_i$, with R the universal gas constant and T the absolute temperature.

Atomistic model: Due to the characteristic size- and time scales of ion transport through nanopores of relevant size, the particle simulations cannot be carried out using all-atom molecular dynamics. Instead, the atomistic system is represented as an overdamped Brownian dynamics simulation, a stochastic particle dynamics system based on the Langevin equation:

$$m_i \frac{d^2 \mathbf{x}_i(t)}{dt^2} = \mathbf{F}_i(\mathbf{x}_i(t)) - m_i \gamma_i \frac{d\mathbf{x}_i(t)}{dt} + \mathbf{R}_i(t), \quad (19)$$

where i is a particle index ($i = 1, 2, \dots, N$), m_i and \mathbf{x}_i are the mass and position, γ_i is the (possibly position-dependent) friction coefficient, \mathbf{F}_i is the systematic force and $\mathbf{R}_i(t)$ is the random force acting on the i th particle at time t .

Overdamped Brownian dynamics are the special case of equation 19 in which the time step dt is sufficiently long that any particle momentum at time t is fully dissipated at time $t + \Delta t$ (equivalent to $\gamma \Delta t \gg 1$). In this case, the Langevin equation can be integrated using a simple explicit scheme [42]

$$\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \mathbf{F}_i(t) \frac{\Delta t}{\gamma_i m_i} + \mathbf{x}_i(\Delta t), \quad (20)$$

where $\mathbf{x}_i(\Delta t)$ is a Gaussian distributed noise with zero mean and a standard deviation $2k_B T \Delta t (m_i \gamma_i)^{-1}$.

The systemic force contribution, $\mathbf{F}_i(t)$ to particle i in solution consists of contributions from an applied electric field (\mathbf{F}_{field}), interactions with charges on the membrane wall ($\mathbf{F}_{membrane}$), and a reaction field describing interaction between particle i and all other particles in solution ($\mathbf{F}_{reaction}$):

$$\begin{aligned} \mathbf{F}_i(t) &= \mathbf{F}_{i,field} + \mathbf{F}_{i,membrane} \\ &\quad + \mathbf{F}_{i,particle-particle} + \mathbf{F}_{i,particle-mesh}. \end{aligned} \quad (21)$$

The BD simulation geometry is shown schematically in Fig. 10. The system is run as a non-equilibrium simulation, with the absolute number of particles constrained within each domain (inlet and outlet) of the simulation system. The nanopore is aligned with the z -axis of the simulation system, and flux is monitored through the xy -plane at the center of the nanopore. At each simulation step, concentration is maintained in the inlet and outlet domains by moving particles between the handshake regions at the edges of the simulation system.

Due to the non-uniform dielectric environment, commonly used accurate ways of accelerating long-range electrostatic calculations for multi-particle systems, such as Particle Mesh Ewald summation and Fast Multipole methods, are not appropriate for particle-particle evaluations. Long-range forces are obtained through direct Coulomb summation, computing forces along the shortest path through the solution phase.

3.3.2. Multiscale implementation

While the solution for this force is simple in a pure atomistic system, the solution here is complicated by the fact that the force contributions are strongly coupled between the atomistic and continuum regions. For a sufficiently small number of particles, the force con-

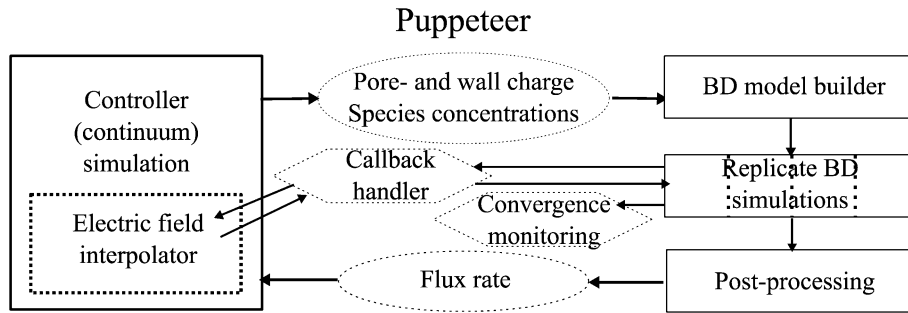


Figure 11. Flow chart for multiscale electrodesalination simulations.

tribution can be obtained by solving all electrostatic interactions at the continuum level, with explicit charges placed at the particle positions in the atomic simulation. This becomes computationally intractable for large numbers of particles or long simulation durations.

In order to reduce the overall computational cost of force evaluations for the particle domain, some core assumptions are made:

- The electric field resulting from the applied (driving) electric potential is computed from an all-continuum solution of the simulation domain, permitting $\mathbf{F}_{i,field}$ to be obtained through $O(N)$ evaluations at the continuum level.
- Membrane charges are restricted to the atomistic domain and assumed to be invariant. This permits $\mathbf{F}_{i,membrane}$ to be pre-computed onto a grid and obtained through Hermite interpolations.

Timescales between the continuum and atomistic models overlap notably more in this simulation than in the preceding simulations of electroosmotic flow. In a fully time integrated scheme, this could be accommodated through projective integration [43], but in the current study the focus was on steady-state transport through the nanopore. The simulation deviates from a traditional Schwarz-decomposition scheme due to the field evaluation callbacks between the BD and continuum components. Since destructive updates to the continuum simulation are not permitted during the BD simulations, these callbacks can be performed in parallel environments where the communication group size for the BD simulation differs from that of the continuum simulation. The added cost in replicating continuum data between nodes is negligible (at most a few megabytes) and eliminates any MPI communication associated with the 10^6 evaluations of external field effects on the BD particles.

As is indicated in Fig. 11, the high-level (python) puppeteer component mediates both the transfer of boundary conditions and field-evaluation callbacks between the continuum and BD components. The performance cost associated with transferring boundary conditions is near-zero (sub-milliseconds for multiple hours of BD simulations), and the total cost imposed for field evaluation callbacks is close to 1% of total simulation time. For this reason, we feel that the improved flexibility and loose coupling obtained by routing the field callbacks through the high-level puppeteer component is well justified.

3.3.3. Results

The multiscale coupling scheme outlined above has been applied to the simulation of ionic fluxes through a nanopore, with a diameter of 5 nm and a length of 50 nm. In the simulations, the three-dimensional Brownian Dynamics (BD) domain covered the full nanopore as well as two hemispheric regions with a 10 nm radius at the entrance and the exit of the pore. The continuum species transport equations were solved on a 2D axisymmetric domain extending from the edge of the BD domain up to 50 nm away from the pore in the streamwise direction, and 25 nm in the radial direction, both upstream and downstream from the pore (see Fig. 12). The Poisson equation for the electric potential was solved on a 2D axisymmetric domain, 150 nm long and 25 nm in the radial direction, containing an axial slice of the pore, the membrane (not shown in Fig. 12) as well as the continuum species transport zones upstream and downstream from the pore. In the fluid, the relative dielectric permittivity was set to 78, while in the membrane, it was set to 2.

Starting from an analytical guess for the species fluxes, the simulations iterate between the continuum and the BD modules until the species flux through the nanopore converges to a steady value. For the computation of the ionic fluxes through the nanopore region

with BD, 10 replica simulations were used, which were run for 10 μs each. Generally, after 2–3 iterations, the changes in the flux values between successive iterations were smaller than the standard deviation of the fluxes across the 10 replicas.

Figure 12 shows the steady state concentration fields for the sodium and chloride ions for the case with a 150 mM NaCl concentration at the domain inlet and 50 mM at the outlet. A 200 mV potential difference was applied across the membrane. The pore surface inside the membrane had a surface charge of -0.01 C/m^2 . The values of the concentration fields in the continuum zone were extracted directly from the Sundance solver. To get the concentration fields inside the nanopore, sampled particle fields from the BD solver were binned and projected onto a mesh on the continuum level. Despite the noise in the sampled con-

centration fields, it is clear that the concentration fields match up well at the interfaces between the continuum and particle models. As can be expected, the bulk of the concentration drop occurs across the nanopore region, with only about 1 mM concentration drop across the continuum entrance and exit regions.

Given the negative charge on the pore surface inside the membrane, the positive ions preferentially migrate towards the pore surface to shield the negative wall charge. As a result, the sodium ion concentration shows a maximum near the pore wall and the chloride ion concentration a minimum. The net result is a positive bulk charge in the fluid near the pore wall, as indicated by Fig. 13, which shows the sum of the concentration fields in Fig. 12. The predicted ionic fluxes through the pore for this case were $8.0 \pm 0.1 \times 10^{-16} \text{ mol/s}$ for the sodium ion and $-5.7 \pm$

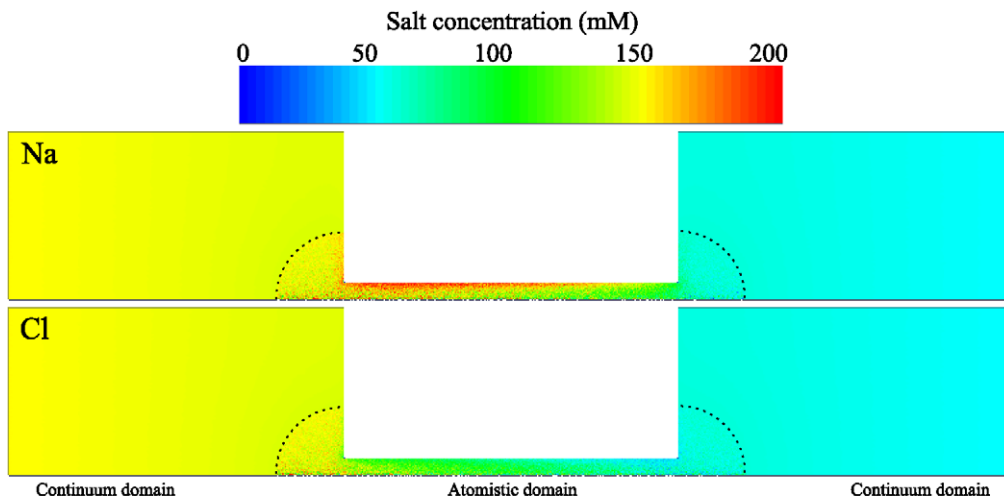


Figure 12. Sodium (top) and chloride (bottom) concentration fields in electro dialysis through a negatively charged pore through a membrane. An axial cross section of the pore is shown in the center, in between a continuum species transport zone upstream and downstream. The membrane material surrounding the pore is not shown. Black dotted lines indicate the location of the handshake region between the continuum and BD models.

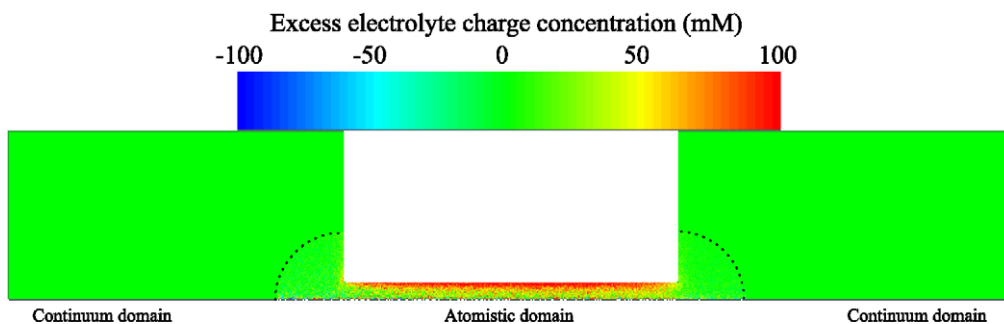


Figure 13. Solution charge density (represented as signed electrolyte concentration) in electro dialysis through a negatively charged nanoporous desalination membrane.

1.6×10^{-17} mol/s for the chloride ion (where a negative flux indicates an upstream flux from right to left).

The particle representation inside the nanopore is essential in these simulations, as the continuum PNP formulation ignores particle diameters, and would therefore pack too many ions too close to the surface in order to shield the negative wall charge. The continuum formulation, on the other hand, is well suited to compute the effect of the externally applied potential difference across the membrane, taking into account the different dielectric permittivities between the membrane and the fluid. The two formulations therefore complement each other well.

The multiscale simulation system described above has further been applied to a parametric study of the effects of membrane surface charge on electrodialysis in nanopores; the results of which will be reported elsewhere.

4. Discussion

The advantages of component-driven design with flexible high-level runtime logic has been demonstrated on two multiscale multiphysics simulations of interface effects in electrokinetic electrolyte transport. For each simulation, the implementation of the coupled multiscale simulation was constructed out of traditional single-physics simulation components bridged with minimal invasion to introduce external interfaces for management of boundary conditions, and modest amounts of high-level Python scripts to handle simulation control and data translation. Even more importantly, the interfaces into the separate simulation components did not need to comply to a common simulation framework, thanks to the dynamic nature of the scripting interface and flexibility afforded by the cross-language polymorphism in the SWIG interface bindings. Given reasonably coarse granularity of data transmission during cross-component callback routines, the performance penalty associated with the introduction of the puppeteer component was far less than one percent of the overall runtime.

These studies have also helped identify areas that hamper the design and implementation of reliable, robust multiscale simulations on parallel simulation platforms. Chief among those are:

- The MPI standard, long the workhorse of parallel computing, is not sufficient to provide the facilities desirable for these simulations. In particular:

- Fault tolerance and error recovery are not mandated by the standard, and are woefully inadequate in its major implementations.
- Support for multithreaded programs is not required by the standard; while the major implementations offer some thread support these features are unreliable and often not supported by administrators of large compute systems.
- Differences in discretization and representation accuracy between atomistic and continuum simulations can cause hard-to-identify convergence problems when iterating between component models. While these problems can be reduced through (vastly) increased sampling, more work is needed on detecting and quantifying the source and magnitude of these errors.

The best features of the component-based simulation design are apparent in the ease of adapting the simulation methods to system configurations and conditions outside of the original design concepts of the initial work. A significant portion of this flexibility originates from the use of the Sundance and PST component toolkits, both of which have fine-grained SWIG bindings, but a large portion also comes from adherence to the design constraints laid out in Section 2. By shifting shared operations such as data transfer and convergence management out to the puppeteer component, behavior of the multiscale coupling could be modified and tested without impinging on the numerically intensive simulation components, and changes in component interfaces could be accommodated through dynamic typing and higher-order functions rather than cascading across program boundaries.

In summary, mixed programming models, in which simulation parameters, simulation control, and high-level runtime logic are unified in a dynamically typed interface (e.g. a scripting language) provide an efficient and highly flexible solution for construction of multiscale multiphysics simulations out of existing simulation programs. By imposing a few modest constraints on the execution order and data sharing, modeling components can be cleanly mapped onto simulation boundaries, providing improved ease and a high degree of flexibility in configuring and adapting the multiscale model for different simulation types. By delegating operations related to the cross-model communications and overall program flow to a dynamically typed, high level programming language, the resulting multiscale simulation system can take advantage of existing and highly optimized simulation software while

providing very loose coupling between the different components and with, in our cases, negligible added computational cost.

Acknowledgements

Work on multiscale simulations of electroosmotic flow was supported by the National Nuclear Security Administration's Advanced Simulation and Computing (ASC) program through Sandia's Computational Science Research Foundation program. Simulations of electro dialysis through nanoporous membranes was supported by the laboratory's program on Water Surety through Laboratory Directed Research and Development funding. Sandia National Laboratories is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the U.S. Department of Energy under contract number DE-AC04-94-AL85000.

References

- [1] A. Warshel and M. Levitt, Theoretical studies of enzymic reactions: Dielectric, electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme, *J. Mol. Biol.* **103**(2) (1976), 227–249.
- [2] N.G. Hadjiconstantinou, Hybrid atomistic-continuum formulations and the moving contact-line problem, *J. Comp. Phys.* **154**(2) (1999), 245–265.
- [3] H.A. Schwarz, Über einige Abbildungsaufgaben, *Journal für Reine und Angewandte Mathematik* **70** (1896), 105–120.
- [4] O. Aktas and N.R. Aluru, A combined continuum/DSMC technique for multiscale analysis of microfluidic filters, *J. Comp. Phys.* **178**(2) (2002), 342–372.
- [5] L. Ren, D. Sinton and D. Li, Numerical simulation of microfluidic injection processes in crossing microchannels, *J. Micromech. Microeng.* **13**(5) (2003), 739–747.
- [6] G. Hu and D. Li, Multiscale phenomena in microfluidics and nanofluidics, *Chem. Eng. Sci.* **62**(13) (2007), 3443–3454.
- [7] K.R. Long, Sundance 2.1, <http://software.sandia.gov/sundance>.
- [8] H. Adalsteinsson, The particle simulation toolkit, to be published.
- [9] SWIG: Simplified Wrapper and Interface Generator, <http://www.swig.org>.
- [10] G. Moy, B. Corry, S. Kuyucak and S.H. Chung, Tests of continuum theories as models of ion channels. I. Poisson–Boltzmann theory versus Brownian dynamics, *Biophys. J.* **78**(5) (2000), 2349–2363.
- [11] B. Corry, S. Kuyucak and S.-H. Chung, Tests of continuum theories as models of ion channels. II. Poisson–Nernst–Planck theory versus Brownian dynamics, *Biophys. J.* **78**(5) (2000), 2364–2381.
- [12] B. Eisenberg, Proteins, channels and crowded ions, *Biophys. Chem.* **100**(1-3) (2003), 507–517.
- [13] A.L. Rollet, M. Jardat, J.F. Dufreche, P. Turq and D. Canet, Multiscale dynamics in ionic media, *J. Mol. Liq.* **92**(1/2) (2001), 53–65.
- [14] S. Succi, O. Filippova, G. Smith and E. Kaxiras, Applying the lattice Boltzmann equation to multiscale fluid problems, *Comput. Sci. Eng.* **3**(6) (2001), 26–37.
- [15] D. Gillespie, W. Nonner and R.S. Eisenberg, Coupling Poisson–Nernst–Planck and density functional theory to calculate ion flux, *J. Phys. Condens. Matter* **14** (2002), 12129–12145.
- [16] S. Koneshan, R.M. Lynden-Bell and J.C. Rasaiah, Friction coefficients of ions in aqueous solution at 25°C, *J. Am. Chem. Soc.* **120**(46) (1998), 12041–12050.
- [17] T.A. van der Straaten, G. Kathawala, A. Trellakis, R.S. Eisenberg and U. Ravaioli, BioMOCA – A Boltzmann transport Monte Carlo model for ion channel simulation, *Mol. Sim.* **31**(2/3) (2005), 151–171.
- [18] S.V. Ermakov, S.C. Jacobson and J.M. Ramsey, Computer simulations of electrokinetic injection techniques in microfluidic devices, *Anal. Chem.* **72** (2000), 3512–3517.
- [19] H.N. Najm, B.J. Debusschere, O.M. Knio, R.R. Ghanem, A. Matta and O.P. Le Maître, Uncertainty quantification in models of microfluid systems, in: *Proc. IMECE'03, 2003 ASME Int. Mech. Eng. Congress*, Washington, DC, 2003, IMECE2003-43240, **154**, pp. 153–162.
- [20] R. Qiao and N.R. Aluru, A compact model for electroosmotic flows in microfluidic devices, *J. Micromech. Microeng.* **12**(5) (2002), 625–635.
- [21] B.J. Kirby and E.F. Hasselbrink, Zeta potential of microfluidic substrates: 1. Theory, experimental techniques, and effects on separations, *Electrophoresis* **25**(2) (2004), 187–202.
- [22] B.J. Kirby and E.F. Hasselbrink, Zeta potential of microfluidic substrates: 2. Data for polymers, *Electrophoresis* **25**(2) (2004), 203–213.
- [23] M.V. Smoluchowski, *Handbuch der Elektrizität und des Magnetismus*, Barth, Leipzig, 1914.
- [24] F.A. Morrison, Electrophoresis of a particle of arbitrary shape, *J. Col. Int. Sci.* **34**(2) (1970), 210–214.
- [25] E. Biddiss, D. Erickson and D.Q. Li, Heterogeneous surface charge enhanced micromixing for electrokinetic flows, *Anal. Chem.* **76**(11) (2004), 3208–3213.
- [26] S. Barany, N.A. Mishchuk and D.C. Prieve, Superfast electrophoresis of conducting dispersed particles, *J. Col. Int. Sci.* **207**(2) (1998), 240–250.
- [27] Y. Yang, K.M. Mayer and J.H. Hafner, Quantitative membrane electrostatics with the atomic force microscope, *Biophys. J.* **92** (2007), 1966–1974.
- [28] R.F. Probstein, *Physicochemical Hydrodynamics: An Introduction*, 2nd edn, Wiley, New York, 1994.
- [29] B.J. Debusschere, H.N. Najm, A. Matta, O.M. Knio, R.G. Ghanem and O.P. Le Maître, Protein labeling reactions in electrochemical microchannel flow: numerical simulation and uncertainty propagation, *Phys. Fluids* **15**(8) (2003), 2238–2250.
- [30] W.F. van Gunsteren, S.R. Billeter, A.A. Eising, P.H. Hünenberger, P.P. Krüger, A.E. Mark, W.R.P. Scott and I.G. Tironi, Biomolecular simulation: The GROMOS96 manual and user guide, 1996.

- [31] H.J.C. Berendsen, J.P.M. Postma, W.F. van Gunsteren, A. Dinola and J.R. Haak, Molecular dynamics with coupling to an external bath, *J. Chem. Phys.* **81**(8) (1984), 3684–3690.
- [32] A.E. Reed, R.B. Weinstock and F. Weinhold, Natural population analysis, *J. Chem. Phys.* **83**(2) (1985), 735–746.
- [33] C.L. Janssen, E.T. Seidl and M.E. Colvin, Object-oriented implementation of parallel ab initio programs, *ACS Symposium Series* **592** (1995), 47–61.
- [34] H.J.C. Berendsen, J.R. Grigera and T.P. Straatsma, The missing term in effective pair potentials, *J. Phys. Chem.* **91**(24) (1987), 6269–6271.
- [35] U. Essmann, L. Perera, M.L. Berkowitz, T. Darden, H. Lee and L.G. Pedersen, A smooth particle mesh Ewald method, *J. Chem. Phys.* **103**(19) (1995), 8577–8593.
- [36] R.D. Braatz, R.C. Alkire, E. Seebauer, E. Rusli, R. Gunawan, T.O. Drews, X. Li and Y. He, Perspectives on the design and control of multiscale systems, *J. Process Control* **16** (2005), 193–204.
- [37] J.E. Miller, Review of water resources and desalination technologies, Sandia Technical Report, SAND2003-0800, 2003.
- [38] Natural Defense Research Council, Energy down the drain – the hidden cost of California’s water supply, 2004 <http://www.nrdc.org/water/conservation/edrain/contents.asp>.
- [39] California Energy Commission, California’s water – energy relationship, Staff Report CEC-700-2005-011-SF, 2005, <http://www.energy.ca.gov/2005publications>.
- [40] P. Aimar, A study of selective electro dialysis and the possibilities offered by coupling with electrolysis, *Int. Chem. Eng.* **25**(2) (1985), 266–274.
- [41] R. Eisenberg and D. Chen, Poisson–Nernst–Planck (PNP) theory of an open ionic channel, *Biophys. J.* **64**(2) (1993), A22.
- [42] W.F. van Gunsteren, H.J.C. Berendsen, J.A.C. Rullmann, Stochastic dynamics for molecules with constraints – Brownian dynamics of *n*-alkanes, *Mol. Phys.* **44**(1) (1981), 69–95.
- [43] C.I. Siettos, M.D. Graham and I.G. Kevrekidis, Coarse Brownian dynamics for nematic liquid crystals: bifurcation, projective integration, and control via stochastic simulation, *J. Chem. Phys.* **118**(22) (2003), 10149–10156.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

