INTEGRAL KERNEL METHODS FOR NONLINEAR PARABOLIC-ELLIPTIC SYSTEMS*

HENRIQUE B. N. MONTEIRO† AND DANIEL M. TARTAKOVSKY‡

Abstract. Nonlinear parabolic-elliptic systems arise in many physical, biological, and chemical phenomena such as chemotaxis, ion transport, self-gravitating particles, and Brownian vortices. Existing methods struggle with the strong coupling and high nonlinearity and nonlocality of some of these systems, especially the ill-conditioned, convection-dominated problems. To overcome numerical difficulties, current approaches rely on initial guesses, preconditioning, or iterative techniques with no convergence guarantees. They might suffer from poor scalability, large memory usage, and difficulty to parallelize. Inspired by the connection of parabolic-elliptic systems to stochastic processes, we introduce a novel meshless, monolithic, and fully explicit method that naturally encapsulates the elliptic and parabolic operators into a single step which updates each node deterministically with global information. By being fully quadrature-based, it avoids solving systems of discretized equations and does not utilize initial guesses or preconditioning, while requiring little memory and being easy to parallelize. We first derive the method in an integral kernel formulation with quadratic complexity in the number of integration nodes and then leverage kernel-independent fast multipole methods (FMM) to present a scalable algorithm with linear complexity. We provide numerical examples for the Poisson-Nernst-Planck equations in one, two, and three dimensions, together with the derivation of the integral kernel for each case. The examples demonstrate the fast convergence and scalability of the FMM-accelerated algorithm, as well as its suitability for convection-dominated problems, making it competitive against traditional PDE solvers.

Key words. multiphysics, nonlinear, PDE systems, PNP equations

MSC codes. 35G61, 35Q81, 35Q92

1. Introduction. A wide range of physical, chemical, and biological phenomena are described by systems of partial differential equations (PDEs). When used in the context of multiphysics and/or multiscale simulations, such systems often comprise coupled PDEs of different types, e.g., deterministic and stochastic or parabolic and elliptic. The coupling between PDEs can be weak or strong [**?**], and their numerical solution starts with a choice of how to handle the interactions between the equations and, for time-dependent problems, of a time-integration strategy. At each time step, one can either evaluate the entire system jointly or solve (possibly with different integrators) each sub-problem separately and perform some synchronization/communication in between time steps [**?**]. The former approach is referred to as *monolithic* or *direct*, while the latter is known as *iterative* or *partitioned*; both can be leveraged in solvers with *loose* or *tight* coupling strategies [**?**].[1]

The direct approach often requires specialized methods or software designed for the system of interest, while the iterative approach is more general and can readily leverage conventional numerical methods originally designed to solve each type of PDE present in the system. This divide-and-conquer strategy often enables iterative

*Submitted to the editors 02/06/2025.

Funding: This work was funded in part by the Air Force Office of Scientific Research under grant FA9550-21-1-0381 and FA9550-24-1-0237, and by the Office of Advanced Scientific Computing Research within the Department of Energy Office of Science under award number DE-SC0023163

†Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA (hbnm@stanford.edu).

‡Institute for Computational and Mathematical Engineering and Department of Energy Science and Engineering, Stanford University, Stanford, CA (tartakovsky@stanford.edu).

[1]We follow [**?**] in referring to strong (weak) coupling of PDEs with respect to the interactions of the physical system, while tight (loose) refers to the degree of interactions between the components of the numerical method.

methods to be faster than solutions to the large, and sometimes ill-conditioned, linear systems constructed by direct approaches. On the other hand, iterative solvers might not converge to the right solution, especially when the coupling causes the system to be highly nonlinear [**?**, **?**].

The method developed here is direct, as it advances the entire PDE system in a single integration step, and relies entirely on quadrature/cubature rules so that it is inherently free from the aforementioned computational difficulties of working with very large and possibly poorly conditioned matrices that arise in linear systems. Unlike most direct methods, ours employs a fully explicit time-stepping scheme and naturally lends itself to fast multipole acceleration for very competitive scalability.

Our method is designed for a class of nonlinear parabolic-elliptic systems that describe many phenomena governed by conservation laws operating on vastly different time scales. They consist, e.g., of a drift-diffusion (parabolic) PDE coupled to a Poisson (elliptic) equation that describes a quantity that affects the dynamics of drift and/or diffusion. Examples of such systems include the Keller-Segel equations for chemotaxis [**?**], the Poisson-Nernst-Planck equations of electrodiffusion [**?**], the Smoluchowski-Poisson equations for self-gravitating Brownian particles [**?**], and systems for two-dimensional Brownian vortices [**?**].

The ubiquity of parabolic-elliptic systems provided impetus to the development of numerical algorithms for their solution. These methods fall into two groups, grid-based approaches and meshless (stochastic particle-based) techniques [**?**, **?**, **?**, **?**]. These methods might target either the original system or a transformed problem with the introduction of an "entropy" variable [**?**, **?**], a Slotboom transformation [**?**], or a (Wasserstein) gradient flow structure [**?**].

Grid-based methods, which compromise the majority of studies on the topic, achieve spatial discretization through finite elements [**?**, **?**, **?**], including (discontinuous) Galerkin methods [**?**, **?**]; finite volumes [**?**, **?**, **?**]; or (generalized) finite differences [**?**, **?**]. The Scharfetter-Gummel scheme [**?**] and its many variants are often employed due to their stability in drift-dominated problems [**?**]. The method's stability guides the selection mostly of implicit or semi-implicit time-discretization methods, but previous work has experimented with a variety of schemes, including explicit [**?**] and semi-implicit [**?**] Euler methods, Adams–Bashforth–Crank–Nicolson method [**?**], and backward differentiation formulae [**?**, **?**], to mention a few.

The systems arising from these discretizations often exhibit considerable nonlinearity and nonlocality [**?**, **?**], motivating the development of bespoke stable, convergent, and structure-preserving algorithms. A representative example of iterative approaches is the Gummel method and related techniques [**?**] that achieve efficiency by decoupling the parabolic and elliptic PDEs, but struggle to ensure stability and convergence in strongly coupled, highly nonlinear systems without modifications such as relaxation or multigrid techniques [**?**]. As an alternative, robust direct methods with guaranteed properties such as mass conservation, unconditional stability, and positivity have been developed for these challenging systems [**?**, **?**, **?**]. These methods require solving large, computationally-expensive nonlinear systems at each time step through either fixed point iteration [**?**] or Newton-type routines, e.g., quasi-Newton [**?**], inexact-Newton [**?**], Newton-Krylov [**?**]. Another downside of these direct methods is their dependence on the quality of both initial guess and preconditioner, with several modifications suggested to assist convergence. These include multiple block preconditioners applied to a single solver [**?**], (algebraic) multigrid methods for preconditioning [**?**], and adaptive grids [**?**] and time steps [**?**] for stability and accuracy. Despite numerous improvements, Newton-type iterations are not guaranteed to

converge for these systems, and fixed point iterations have only been shown to be convergent in limited scenarios (see [**?**] and the references therein).

We introduce a new direct method that does not require as a subroutine the solution of any system of equations, completely eliminating the need for initial guesses, preconditioning, and solvers for large matrices. It is based on an integral kernel reformulation of the PDE system and has the advantage of being fully explicit, an unusual feature among competing direct methods, due to its usage of explicit quadrature/cubature rules in lieu of large systems of equations.[2] Our method is also inherently robust for ill-conditioned, convection-dominated problems due to the usage of global information for time stepping. Each node is updated using information from every node at some previous time, allowing stability in much larger time steps. This strong property of global information updates is absent in the local discretizations of the aforementioned approaches since they would yield computationally prohibitive, large, dense systems. In contrast, we demonstrate linear time complexity in the number of update nodes thanks to kernel-independent fast multipole [**?**, **?**] acceleration.

We provide a formal statement of the problem in section 2 and develop a general (agnostic of initial and boundary conditions) methodology for its solution in section 3. In section 4, this methodology is used to solve the Poisson-Nernst-Planck system in one, two, and three spatial dimensions. In section 5, we discuss the advantages and limitations of our approach, as well as an outlook of future work.

2. Problem formulation. A system of two coupled PDEs describes the spatiotemporal evolution of state variables $\varphi(\mathbf{x}, t)$ and $p(\mathbf{x}, t)$ within d-dimensional domain $\Omega \subseteq \mathbb{R}^d$ at time $t \in \mathbb{R}^+$. The first PDE involves a linear elliptic spatial differential operator $\mathcal{L}$,

$$\mathcal{L}(\varphi) = \lambda p, \tag{2.1a}$$

where $\lambda \neq 0$ is a real constant. The second is a (possibly unnormalized) Fokker-Planck (FP) equation,

$$\frac{\partial p}{\partial t} = \sum_{i=1}^{d} \sum_{j=1}^{d} \frac{\partial^2 (D_{ij} p)}{\partial x_i \partial x_j} - \sum_{i=1}^{d} \frac{\partial (\mu_i p)}{\partial x_i}, \tag{2.1b}$$

in which $\mu_i(\mathbf{x}, t; \{\partial^\alpha \varphi\})$ and $D_{ij}(\mathbf{x}, t; \{\partial^\alpha \varphi\})$ are the components of drift velocity $\boldsymbol{\mu}$ and diffusion tensor $\mathbf{D}$, respectively; and $\{\partial^\alpha \varphi\}$ denotes, in multi-index notation, an arbitrary set of derivatives of φ. The dependence of $\boldsymbol{\mu}$ and $\mathbf{D}$ on φ and its derivatives introduces the nonlinear coupling of the PDEs, as both need to be advanced together in time for a correct modeling of the dynamics. In many problems of interest, the operator $\mathcal{L}$ in (2.1a) is the Laplacian, $\varphi(\mathbf{x}, t)$ is the electrostatic or gravitational potential, and Eq. (2.1b) is a continuity equation describing mass transport. In such cases, the drift is a function of the gradient $\nabla\varphi$, as the electric or gravitational field influence the movement of mass.

The system evolves from its known state $p_0(\mathbf{x})$ at time $t = 0$, giving rise to the initial condition

$$p(\mathbf{x}, 0) = p_0(\mathbf{x}), \qquad \mathbf{x} \in \Omega; \tag{2.1c}$$

[2] Integral methods like ours are commonly used to solve elliptic problems, undergirding the classical techniques of potential theory [**?**, **?**] and boundary element methods [**?**, **?**]. Yet, to the best of our knowledge, no such approach has ever been developed to solve a nonlinear parabolic-elliptic coupled system.

the corresponding initial distribution of the second state variable, $\varphi(\mathbf{x}, t)$, is a solution of (2.1a) with $p = p_0$. Equations (2.1a) and (2.1b) are also subject to boundary conditions

$$\mathcal{B}_1(\varphi) = g, \quad \mathcal{B}_2(p) = h, \quad \mathbf{x} \in \partial\Omega; \tag{2.1d}$$

prescribed on the closure $\partial\Omega$ of the computational domain Ω. Here, $g(\mathbf{x}, t)$ and $h(\mathbf{x}, t)$ are given functions, and $\mathcal{B}_1$ and $\mathcal{B}_2$ are differential operators specifying the (Dirichlet, Neumann, or Robin) boundary condition.

3. Methodology. Numerical solution of coupled PDE systems, such as (2.1), usually involves operator splitting or another method to alternate between solving (2.1a) and (2.1b) separately at each time step. Our approach advances the system with a single update at each time step, without explicit operator splitting.

3.1. Connection to stochastic processes. Although Fokker-Planck (FP) equations are typically discussed in the context of stochastic processes, their functional form is sufficiently general to encompass a large group of parabolic PDEs, regardless of whether these PDEs are originally derived to describe a physical stochastic process. The solution to the FP equation (2.1b), $p(\mathbf{x}, t)$, is the (possibly unnormalized) single-point probability density function (PDF) of some Itô process $\mathbf{X}(t)$ driven by the following stochastic differential equation (SDE) [**?**],

$$\mathrm{d}\mathbf{X} = \boldsymbol{\mu}(\mathbf{X}, t)\mathrm{d}t + \boldsymbol{\sigma}(\mathbf{X}, t)\mathrm{d}\mathbf{W}, \tag{3.1a}$$

where $\mathbf{W}(t)$ is a d-dimensional Wiener process. The drift vector $\boldsymbol{\mu}$ has components μ_i and the $d \times d$ diffusion matrix $\boldsymbol{\sigma}(\mathbf{x}, t)$ is defined to match the components D_{ij} of the diffusion tensor $\mathbf{D}$ in (2.1b) such that

$$\mathbf{D} = \frac{1}{2}\boldsymbol{\sigma}\boldsymbol{\sigma}^\top. \tag{3.1b}$$

Definition 3.1 (Transition probability density of $\mathbf{X}$). *Given the Itô process $\mathbf{X}(t)$ in (3.1), with the single-point PDF $p(\mathbf{x}, t)$ in (2.1b), the transition PDF of the process from time t' to time t ($t > t'$), $q(\mathbf{x}, t; \mathbf{x}', t')$, is defined as*[3]

$$p(\mathbf{x}, t) = \int_\Omega q(\mathbf{x}, t; \mathbf{x}', t')p(\mathbf{x}', t')\mathrm{d}\mathbf{x}'. \tag{3.2}$$

3.2. Integral kernel formulation. While the system of coupled PDEs in (2.1) is nonlinear, the differential operator $\mathcal{L}$ in (2.1a) is linear. Consequently, it has an associated time-invariant Green's function $G(\mathbf{x}, \tilde{\mathbf{x}})$ defined as a solution of the boundary-value problem

$$\mathcal{L}(G) = \delta(\mathbf{x} - \tilde{\mathbf{x}}), \quad \{\mathbf{x}, \tilde{\mathbf{x}}\} \in \Omega; \qquad \mathcal{B}_1(G) = 0, \quad \mathbf{x} \in \partial\Omega.$$

The Green's function's analogue for the nonlinear system (2.1) is an update kernel K, which is akin to a convolution of the fundamental solutions of each differential operator.

Definition 3.2 (Transition kernel K). *Given the transition PDF of the Itô process $\mathbf{X}(t)$, $q(\tilde{\mathbf{x}}, t; \mathbf{x}', t')$, and the Green's function G for the differential operator $\mathcal{L}$, we define the transition kernel K,*

$$K(\mathbf{x}, t; \mathbf{x}', t') = \int_\Omega G(\mathbf{x}, \tilde{\mathbf{x}})q(\tilde{\mathbf{x}}, t; \mathbf{x}', t')\mathrm{d}\tilde{\mathbf{x}}. \tag{3.3}$$

[3]The density $p(\mathbf{x}, t)$ is independent of t' since q is the true transition density of the process $\mathbf{X}(t)$.

THEOREM 3.3. *Let $\varphi_{\mathrm{b}}(\mathbf{x})$ denote a solution to the homogeneous version of* (2.1a)*, $\mathcal{L}(\varphi) = 0$, subject to the boundary condition $\mathcal{B}_1(\varphi) = g$ from* (2.1d)*. If the product $(Gqp)(\mathbf{x}', \tilde{\mathbf{x}}, \mathbf{x}, t, t')$ is absolutely integrable over $(\mathbf{x}', \tilde{\mathbf{x}}) \in \Omega \times \Omega$ for every $\mathbf{x} \in \Omega$ and $t > t'$, then a solution of the nonlinear PDE system* (2.1) *at time t is related to the solution at earlier time $t' \geq 0$ $(t > t')$ by*

$$\varphi(\mathbf{x}, t) = \int_{\Omega} K(\mathbf{x}, t; \mathbf{x}', t') \mathcal{L}'(\varphi(\mathbf{x}', t')) \mathrm{d}\mathbf{x}' + \varphi_{\mathrm{b}}(\mathbf{x}), \tag{3.4a}$$

$$p(\mathbf{x}, t) = \lambda^{-1} \mathcal{L}(\varphi(\mathbf{x}, t)). \tag{3.4b}$$

The differential operator $\mathcal{L}'$ is w.r.t. $\mathbf{x}'$, while $\mathcal{L}$ is w.r.t. $\mathbf{x}$. Similarly to (3.2)*, the solution does not depend on the arbitrary time t', as clarified in the proof below.*

Proof. Due to the principle of superposition, a solution of (2.1a) and (2.1d) is written in terms of the Green's function G as

$$\varphi(\mathbf{x}, t) = \int_{\Omega} G(\mathbf{x}, \tilde{\mathbf{x}}) \left(\lambda p(\tilde{\mathbf{x}}, t)\right) \mathrm{d}\tilde{\mathbf{x}} + \varphi_{\mathrm{b}}(\mathbf{x}).$$

Accounting for (3.2),

$$\varphi(\mathbf{x}, t) = \int_{\Omega} G(\mathbf{x}, \tilde{\mathbf{x}}) \left(\lambda \int_{\Omega} q(\tilde{\mathbf{x}}, t; \mathbf{x}', t') p(\mathbf{x}', t') \mathrm{d}\mathbf{x}' \right) \mathrm{d}\tilde{\mathbf{x}} + \varphi_{\mathrm{b}}(\mathbf{x}).$$

The absolute integrability of $(Gqp)(\mathbf{x}', \tilde{\mathbf{x}}, \mathbf{x}, t, t')$ allows one to change the order of integration in accordance with Fubini's theorem,

$$\varphi(\mathbf{x}, t) = \int_{\Omega} \left(\int_{\Omega} G(\mathbf{x}, \tilde{\mathbf{x}}) q(\tilde{\mathbf{x}}, t; \mathbf{x}', t') \mathrm{d}\tilde{\mathbf{x}} \right) \left(\lambda p(\mathbf{x}', t')\right) \mathrm{d}\mathbf{x}' + \varphi_{\mathrm{b}}(\mathbf{x}).$$

Writing this result in terms of the transition kernel K from (3.3) yields (3.4a). Finally, (3.4b) results from applying the linear operator $\mathcal{L}$ to (3.4a) according to (2.1a). Note that the proof follows regardless of the choice of t' as long as $t > t'$, so the solution does not depend on it. ∎

Remark 3.4. For the transition PDF $q(\mathbf{x}, t; \mathbf{x}', t') = \delta(\mathbf{x} - \mathbf{x}')$, the transition kernel K becomes the Green's function G, and the statement of Theorem 3.3 degenerates to the first line of the proof. This is the case if $t = t'$ or whenever p has no time structure, i.e., $p(\mathbf{x}, t) = p(\mathbf{x}, t')$ for any $t > t'$. This highlights how the transition kernel K generalizes the Green's function G from operator $\mathcal{L}$ to the full system by incorporating the time dependence of φ.

3.3. Re-parametrization of transition kernel. Given the initial density p_0 in (2.1c) and a time discretization of the simulation time horizon, our goal is to advance the solution for each time step using the update recurrence from Theorem 3.3, which requires the knowledge of the transition kernel K. However, a closed-form analytical expression for K is generally unavailable. Moreover, since K is a mapping from $\mathbb{R}^{2d+2}$ to $\mathbb{R}$, its approximation in some numerical procedure can become computationally prohibitive due to the curse of dimensionality if done naively.

A full precomputation of K is unfeasible since, for times $t_1 < t_2 < t_3$, there is no knowledge of $q(\mathbf{x}, t_3; \mathbf{x}', t_2)$ at time t_1 to approximate the transition kernel $K(\mathbf{x}, t_3; \mathbf{x}', t_2)$. Alternatively, an approximation of K at each time step using only

information available up to that time can be constructed as a simpler mapping from $\mathbb{R}^{2d}$ to $\mathbb{R}$, since we would fix t and t' at each time step. However, the procedure needs to be repeated every time step, so the computational cost might be prohibitive compared to the traditional methods such as finite element and finite volume.

In lieu of either approach, we leverage the connection of the Fokker-Planck to stochastic processes to seek a re-parametrization of K that yields a way to precompute the update kernel K efficiently.

3.3.1. Euler-Maruyama connection. Let $\{t_i\}_{i=0}^{N_{\text{st}}}$ denote a time discretization of the simulation time horizon, from $t_0 = 0$ until $t_{N_{\text{st}}} > 0$, with uniform time step $\Delta t = t_{i+1} - t_i$. The Itô process $\mathbf{X}(t)$ in Eq. (3.1), discretized on this time grid, is a collection of $N_{\text{st}} + 1$ random variables $\mathbf{X}_i \equiv \mathbf{X}(t_i)$, for $i = 0, \ldots, N_{\text{st}}$. Given the initial state of the system, $\mathbf{X}_0$, its subsequent evolution at the jth time step computed through the Euler-Maruyama (EM) method is such that the *tentative* updated state $\tilde{\mathbf{X}}_{j+1}$ (before application of boundary conditions) is

$$\tilde{\mathbf{X}}_{j+1} = \mathbf{X}_j + \boldsymbol{\mu}(\mathbf{X}_j, t_j)\Delta t + \sqrt{2\mathbf{D}(\mathbf{X}_j, t_j)}\Delta \mathbf{W}_j, \tag{3.5}$$

where $\Delta \mathbf{W}_j \equiv \Delta \mathbf{W}(t_j) \sim \mathcal{N}(\mathbf{0}, \Delta t\, \mathbf{I}_d)$ and $\mathbf{I}_d$ denotes the $d \times d$ identity matrix. Since the increments of the Wiener process $\Delta \mathbf{W}_j$ are Gaussian, so is the tentative update of the Itô process $\mathbf{X}_{j+1}$ in (3.5) [**?**, **?**],

$$\tilde{\mathbf{X}}_{j+1} \sim \mathcal{N}(\mathbf{m}^{\Delta t}(\mathbf{X}_j, t_j),\ \mathbf{V}^{\Delta t}(\mathbf{X}_j, t_j)). \tag{3.6}$$

The mean, $\mathbf{m}^{\Delta t} \in \mathbb{R}^d$, and covariance, $\mathbf{V}^{\Delta t} \in \mathbb{R}^{d \times d}$, of this Gaussian distribution are computed as

$$\mathbf{m}^{\Delta t}(\mathbf{x}, t) = \mathbf{x} + \boldsymbol{\mu}(\mathbf{x}, t)\Delta t \tag{3.7a}$$

$$\mathbf{V}^{\Delta t}(\mathbf{x}, t) = 2\mathbf{D}(\mathbf{x}, t)\Delta t. \tag{3.7b}$$

The Gaussianity of the tentative update $\tilde{\mathbf{X}}_{j+1}$ is a property of the EM discretization (3.5). The transition density q of the true continuous process $\mathbf{X}(t)$ is not necessarily Gaussian because of the time- and space-varying characteristics, but the EM scheme is an accurate approximation for sufficiently small time steps.[4] It allows parametrization of the otherwise intractable transition density q in Definition 3.1 with only two quantities, mean and variance.

3.3.2. EM with general boundary conditions. Let $\phi(\mathbf{x}; \boldsymbol{\mathcal{M}}, \boldsymbol{\mathcal{V}})$ denote the multivariate Gaussian PDF with mean $\boldsymbol{\mathcal{M}}$ and covariance $\boldsymbol{\mathcal{V}}$. To incorporate the boundary conditions, if any, the *actual* updated position $\mathbf{X}_{j+1}$ from the EM method is sampled from the transition probability density $q^{\Delta t}$,

$$q^{\Delta t}(\mathbf{x}, t_{j+1}; \mathbf{X}_j, t_j) = \phi(\mathbf{x}; \mathbf{m}^{\Delta t}(\mathbf{X}_j, t_j), \mathbf{V}^{\Delta t}(\mathbf{X}_j, t_j)) + f^{\Delta t}(\mathbf{x}; \mathbf{X}_j). \tag{3.8}$$

The distribution $f^{\Delta t}$ (not necessarily a PDF) is constructed such that $q^{\Delta t}$ is a valid PDF that satisfies the boundary conditions (2.1d). In the absence of boundary conditions, e.g., when the simulation domain is infinite, $f^{\Delta t} = 0$ and $\mathbf{X}_{j+1} = \tilde{\mathbf{X}}_{j+1}$.[5]

[4] See [**?**, **?**] for a discussion on the weak and strong convergence of the EM scheme.

[5] Section 4 contains several examples of transition densities that satisfy particular boundary conditions in different domains.

3.3.3. Gaussian mixture transition densities. The next step in precomputing an approximation of kernel K is to parametrize the EM transition density $q^{\Delta t}$ with a Gaussian mixture $\hat{q}^{\Delta t}$,

$$\hat{q}^{\Delta t}(\mathbf{x}, t_{j+1}; \mathbf{X}_j, t_j) = \sum_{i=1}^{M} c_i \phi(\mathbf{x}; \boldsymbol{\mathcal{M}}_i^{\Delta t}(\mathbf{X}_j, t_j), \boldsymbol{\mathcal{V}}_i^{\Delta t}(\mathbf{X}_j, t_j)), \tag{3.9}$$

in which $\{c_i\}_{i=1}^M$ are real constants, M is the finite number of mixture components, and $\{\boldsymbol{\mathcal{M}}_i^{\Delta t}\}_{i=1}^M$ and $\{\boldsymbol{\mathcal{V}}_i^{\Delta t}\}_{i=1}^M$ are respectively real-valued functions denoting the mean and covariance of each component of the mixture.

This choice of parametrization offers several advantages. First, in the absence of boundary conditions to be enforced, it is an exact representation (with $M = 1$) of the EM transition density in (3.8). For all others cases, when $f^{\Delta t} \neq 0$, even if an exact representation is impossible, an approximation may still be constructed to arbitrary precision since the family of Gaussian mixtures is a universal approximator of continuous densities [**?**, **?**, **?**].

DEFINITION 3.5 (Gaussian transition kernel K^ϕ). *The transition kernel K with Gaussian transition probability, $q(\tilde{\mathbf{x}}, t; \mathbf{x}', t') = \phi(\tilde{\mathbf{x}}; \boldsymbol{\mathcal{M}}^{\Delta t}(\mathbf{x}', t'), \boldsymbol{\mathcal{V}}^{\Delta t}(\mathbf{x}', t'))$ with $\Delta t = t - t'$, is referred to as the Gaussian transition kernel K^ϕ,*

$$K^\phi(\mathbf{x}, t; \mathbf{x}', t') = \int_\Omega G(\mathbf{x}, \tilde{\mathbf{x}}) \phi(\tilde{\mathbf{x}}; \boldsymbol{\mathcal{M}}^{\Delta t}(\mathbf{x}', t'), \boldsymbol{\mathcal{V}}^{\Delta t}(\mathbf{x}', t')) \mathrm{d}\tilde{\mathbf{x}}. \tag{3.10}$$

An alternative parametrization of the Gaussian transition kernel,

$$K^\phi(\mathbf{x}; \boldsymbol{\mathcal{M}}, \boldsymbol{\mathcal{V}}) = \int_\Omega G(\mathbf{x}, \tilde{\mathbf{x}}) \phi(\tilde{\mathbf{x}}; \boldsymbol{\mathcal{M}}, \boldsymbol{\mathcal{V}}) \mathrm{d}\tilde{\mathbf{x}}, \tag{3.11}$$

translates the dependence on the current space and time, $(\mathbf{x}', t')$, into the dependence on statistical moments $(\boldsymbol{\mathcal{M}}, \boldsymbol{\mathcal{V}})$ of the underlying stochastic process under the EM discretization at that particular time and location.

The Gaussian transition kernel K^ϕ provides a means for construction of an approximate update kernel $\hat{K}^{\Delta t}$ as a mixture of Gaussian transition kernels,

$$\hat{K}^{\Delta t}(\mathbf{x}; \mathbf{x}', t') = \sum_{i=1}^{M} c_i K^\phi(\mathbf{x}; \boldsymbol{\mathcal{M}}_i^{\Delta t}(\mathbf{x}', t'), \boldsymbol{\mathcal{V}}_i^{\Delta t}(\mathbf{x}', t')), \quad \Delta t = t - t'. \tag{3.12}$$

The following theorem demonstrates that using the Gaussian mixture transition density $\hat{q}^{\Delta t}$, the update kernel K becomes the approximate update kernel $\hat{K}^{\Delta t}$.

THEOREM 3.6. *If the transition PDF $q(\tilde{\mathbf{x}}, t; \mathbf{x}', t')$ of the Itô process $\mathbf{X}(t)$ in (3.1) is approximated with its Gaussian mixture counterpart $\hat{q}^{\Delta t}(\tilde{\mathbf{x}}, t; \mathbf{x}', t')$, then the update kernel K in (3.3) takes the form of the approximate update kernel $\hat{K}^{\Delta t}$ in (3.12).*

Proof. Setting $q(\tilde{\mathbf{x}}, t; \mathbf{x}', t') = \hat{q}^{\Delta t}(\tilde{\mathbf{x}}, t; \mathbf{x}', t')$ in (3.3) yields

$$K(\mathbf{x}, t; \mathbf{x}', t')_{|_{q=\hat{q}^{\Delta t}}} = \int_\Omega G(\mathbf{x}, \tilde{\mathbf{x}}) \hat{q}^{\Delta t}(\tilde{\mathbf{x}}, t; \mathbf{x}', t') \mathrm{d}\tilde{\mathbf{x}}.$$

According to the definition of $\hat{q}^{\Delta t}(\tilde{\mathbf{x}}, t; \mathbf{x}', t')$ in (3.9),

$$K(\mathbf{x}, t; \mathbf{x}', t')_{|_{q=\hat{q}^{\Delta t}}} = \int_\Omega G(\mathbf{x}, \tilde{\mathbf{x}}) \left[\sum_{i=1}^{M} c_i \phi(\tilde{\mathbf{x}}; \boldsymbol{\mathcal{M}}_i^{\Delta t}(\mathbf{x}', t'), \boldsymbol{\mathcal{V}}_i^{\Delta t}(\mathbf{x}', t')) \right] \mathrm{d}\tilde{\mathbf{x}}$$
$$= \sum_{i=1}^{M} c_i \left[\int_\Omega G(\mathbf{x}, \tilde{\mathbf{x}}) \phi(\tilde{\mathbf{x}}; \boldsymbol{\mathcal{M}}_i^{\Delta t}(\mathbf{x}', t'), \boldsymbol{\mathcal{V}}_i^{\Delta t}(\mathbf{x}', t')) \mathrm{d}\tilde{\mathbf{x}} \right].$$

The theorem is proved by accounting for the definitions of K^ϕ in (3.10) and $\hat{K}^{\Delta t}$ in (3.12). ∎

Theorem 3.6 states that every space-time point $(\mathbf{x}', t')$ can be mapped onto parameters $(\boldsymbol{\mathcal{M}}_i^{\Delta t}, \boldsymbol{\mathcal{V}}_i^{\Delta t})$ of the Gaussian mixture representing the transition probability of stochastic process $\mathbf{X}(t)$ under the EM discretization. This implies that even though kernel K is not known a priori, it can be mapped onto a mixture of Gaussian transition kernels K^ϕ, which can be either explicitly derived or readily precomputed from (3.11).

3.4. Numerical procedure. The first stage of our numerical procedure is to construct the Gaussian mixture $\hat{q}^{\Delta t}$ in (3.9) by choosing the constants $\{c_i\}_{i=1}^M$ and real-valued functions $\{\boldsymbol{\mathcal{M}}_i^{\Delta t}\}_{i=1}^M$ and $\{\boldsymbol{\mathcal{V}}_i^{\Delta t}\}_{i=1}^M$. This procedure, which depends on the domain geometry and boundary conditions, is carried out in section 4 for several computational examples.

The second stage is to derive, or precompute, the Gaussian transition kernel $K^\phi(\mathbf{x}; \boldsymbol{\mathcal{M}}, \boldsymbol{\mathcal{V}})$ in (3.11) in terms of an arbitrary mean, $\boldsymbol{\mathcal{M}}$, and variance, $\boldsymbol{\mathcal{V}}$, of the normal distribution. We derive explicit expressions for K^ϕ for the one-, two-, and three-dimensional Laplace operators in section 4; expressions for other elliptic operators were not pursued but are feasible.

The third stage is to assemble the approximate update kernel $\hat{K}^{\Delta t}$ in (3.12) from $\hat{q}^{\Delta t}$ and K^ϕ. The final stage is to apply Theorem 3.3 to design a numerical method to time step the state variable $\varphi(\mathbf{x}, t)$ through numerical integration,

$$\varphi(\mathbf{x}, t_{j+1}) = \int_\Omega \hat{K}^{\Delta t}(\mathbf{x}; \mathbf{x}', t_j) \mathcal{L}'(\varphi(\mathbf{x}', t_j)) \mathrm{d}\mathbf{x}' + \varphi_{\mathrm{b}}(\mathbf{x}), \tag{3.13}$$

for the time discretization $\{t_i\}_{i=0}^{N_{\mathrm{st}}}$.

At each time step, the state variable $\varphi(\mathbf{x}, \cdot)$ is evaluated at a set $S = \{\mathbf{x}_i\}_{i=1}^{N_{\mathrm{pt}}} \in \Omega$ of N_{pt} points for the numerical integration in (3.13). To simplify the presentation, we assume that $\varphi(\mathbf{x}, \cdot)$ is evaluated on the same set S for all time steps. Then, for $\mathbf{x}_k \in S$, Eq. (3.13) is numerically evaluated as

$$\varphi(\mathbf{x}_k, t_{j+1}) = \sum_{i=1}^{N_{\mathrm{pt}}} w_i \cdot \hat{K}^{\Delta t}(\mathbf{x}_k; \mathbf{x}_i, t_j) \cdot \mathcal{L}_i(\varphi(\mathbf{x}_i, t_j)) + \varphi_{\mathrm{b}}(\mathbf{x}_k), \tag{3.14}$$

where $\mathcal{L}_i$ denotes the differential operator with respect to $\mathbf{x}_i$, and the dot indicating the scalar multiplication is used for readability.

The set of points S and the corresponding set of integration weights $\{w_i\}_{i=1}^{N_{\mathrm{pt}}}$ are determined by a chosen quadrature/cubature rule (see [**?**, **?**, **?**, **?**]).[6] Regardless of this choice, a final algorithm has complexity $\mathcal{O}(N_{\mathrm{pt}}^2 N_{\mathrm{st}})$ since, at each of the N_{st} time

[6] In section 4, we employ several of these methods for each computational example.

steps, the evaluation of (3.14) involves a summation of $N_{\rm pt}$ terms for each of the $N_{\rm pt}$ points in S.

At each step, we need to compute $\mathcal{L}_i(\varphi(\mathbf{x}_i, t_j))$ to evaluate (3.14). Since it is the value of the elliptic operator applied to the previous time solution, it is computable directly from $\varphi(\cdot, t_j)$ using numerical differentiation. Instead, we apply the operator $\mathcal{L}_k$ to both sides of that equation to derive a recursive update formula,

$$\mathcal{L}_k\varphi(\mathbf{x}_k, t_{j+1}) = \sum_{i=1}^{N_{\rm pt}} w_i \cdot \mathcal{L}_k(\hat{K}^{\Delta t}(\mathbf{x}_k; \mathbf{x}_i, t_j)) \cdot \mathcal{L}_i(\varphi(\mathbf{x}_i, t_j)) + \mathcal{L}_k(\varphi_{\rm b}(\mathbf{x}_k)). \tag{3.15}$$

Direct evaluation of (3.15) is computationally cheaper than numerical differentiation because of the availability of explicit expressions for $\mathcal{L}K^\phi$ and, thus, $\mathcal{L}\hat{K}^{\Delta t}$ from (3.12). To illustrate the advantage of (3.15), consider a second-order centered difference scheme for the second derivative of some function f, $f''(x) \approx [f(x-h) - 2f(x) + f(x+h)]/h^2$ for a small $h > 0$, to compute the Laplacian ($\mathcal{L} = \nabla^2$) of φ at $\mathbf{x}_k$ in three spatial dimensions. Function φ must be evaluated at six locations in addition to $\mathbf{x}_k$, with each evaluation requiring a summation over $N_{\rm pt}$ terms per (3.14), as opposed to the alternative of evaluating a single sum of $N_{\rm pt}$ terms in (3.15).

Algorithm 3.1 Integral method for parabolic-elliptic systems

Input: Time discretization $\{t_i\}_{i=0}^{N_{\rm st}}$, quadrature with evaluation set $S = \{\mathbf{x}_k\}_{k=1}^{N_{\rm pt}}$ and weights $\{w_k\}_{k=1}^{N_{\rm pt}}$, and approximate update kernel $\hat{K}^{\Delta t}(\mathbf{x}; \mathbf{x}', t')$ (3.12).
Output: $\varphi(\mathbf{x}_k, t_{N_{\rm st}})$ and $p(\mathbf{x}_k, t_{N_{\rm st}})$ for every $\mathbf{x}_k \in S$.

```
Set Lφ(x_k, t_0) = λ p_0(x_k) for every x_k ∈ S
for t = t_1, ..., t_{N_st} do
    for each x_k ∈ S do
        Compute φ(x_k, t) from Eq. (3.14)
        Compute Lφ(x_k, t) from Eq. (3.15)
    end for
end for
Set p(x_k, t_{N_st}) = λ^{-1} Lφ(x_k, t_{N_st}) for every x_k ∈ S
```

Algorithm 3.1 combines the various components of our fully explicit and direct numerical procedure for solving the parabolic-elliptic systems (2.1). It commences by choosing a time discretization and a quadrature/cubature rule, and by defining the approximate update kernel $\hat{K}^{\Delta t}$. The latter can be accomplished either by constructing the kernel directly, as we do in the two- and three-dimensional examples below, or by specifying the Gaussian mixture $\hat{q}^{\Delta t}$ in (3.9) and the Gaussian transition kernel K^ϕ in (3.11) to indirectly define $\hat{K}^{\Delta t}$ through (3.12). That is the procedure we detailed above and used in the one-dimensional example.

This initialization procedure is followed by evaluating $\mathcal{L}\varphi(\cdot, t_0) = \lambda p_0$ with p_0 specified by the initial condition (2.1c), computing $\varphi(\cdot, t_1)$ from (3.14) at every integration node, and evaluating $\mathcal{L}\varphi(\cdot, t_1)$ via (3.15). The repeated evaluation of (3.14) and (3.15) advances the problem further in time until termination. We record the solution at the final time, but it is available at each simulation time step.

In some problems, the gradient or other derivatives of φ may be required to evaluate $\hat{K}^{\Delta t}$ if $\hat{q}^{\Delta t}$ depends on them.[7] These cases require the evaluation of an

[7] That happens in the Poisson-Nernst-Planck system (section 4), which requires knowledge of $\nabla\varphi$.

additional expression of the same form as (3.15), with $\mathcal{L}_k$ swapped by the relevant linear differential operator, e.g., ∇, to advance in time the necessary derivatives while avoiding numerical differentiation for the aforementioned reasons. This generalization requires the evaluation of one more update expression in addition to (3.14) and (3.15) at each time step, i.e., adds one more line inside the inner **for** loop in Algorithm 3.1.

3.5. FMM-accelerated algorithm. The core idea behind the algorithm, grounded in Theorem 3.3 and expressed in Eq. (3.13), does not immediately resemble an N-body problem. Set S defines the integration nodes rather than a collection of interacting particles, and, unlike particles, it is assumed to be fixed for every time step. Yet, as we demonstrate next, the numerical integration routine can be recast as an N-body problem over an interacting kernel in a manner agnostic to the quadrature method. In this novel interpretation, the application of kernel-independent fast multipole methods (FMM) [**?**, **?**] is trivial. That decreases the complexity of the overall algorithm from $\mathcal{O}(N_{\text{pt}}^2 N_{\text{st}})$ to $\mathcal{O}(N_{\text{pt}} N_{\text{st}})$, making it particularly competitive against traditional PDE solvers, as we demonstrate in section 4.

The key insight comes from the realization that the summation in (3.14) is analogous to an N-body problem amenable to FMM,

$$\sum_{i=1}^{N_{\text{pt}}} \underbrace{w_i \cdot \mathcal{L}_i(\varphi(\mathbf{x}_i, t_j))}_{\text{FMM weights}} \cdot \underbrace{\hat{K}^{\Delta t}(\mathbf{x}_k; \mathbf{x}_i, t_j)}_{\text{FMM kernel}}, \tag{3.16}$$

since it is evaluated for every $\mathbf{x}_k \in S$ at each time step. Each point influences every other point, exactly as in an N-body problem. With the FMM weights and kernel thus identified, any suitable off-the-shelf implementation of a kernel-independent FMM method is readily applicable. In section 4, we utilize the FMM package `BBFMM2D` to demonstrate the FMM acceleration in a two-dimensional example, and we refer the reader to the original paper [**?**] for details of the method. Algorithm 3.1 remains mostly unaffected by the FMM acceleration, the sole modification being the replacement of its inner loop (lines 3-5) with the FMM evaluation of (3.14) and (3.15).

4. Application to PNP equations. We use the Poisson-Nernst-Planck (PNP) equations for a single ionic species, a special case of the general parabolic-elliptic PDE system (2.1), to illustrate the salient features of the proposed algorithm. These equations codify conservation of electric charge and mass, describing the spatiotemporal evolution of electrostatic potential $\varphi(\mathbf{x}, t)$ and ion concentration $c(\mathbf{x}, t)$,

$$\nabla^2 \varphi = -\frac{zF}{\epsilon_0 \epsilon_{\text{r}}} c, \tag{4.1a}$$

$$\frac{\partial c}{\partial t} = -\nabla \cdot \mathbf{J}. \tag{4.1b}$$

Here, z is the valence of the ions, F is the Faraday constant, ϵ_0 is the permittivity of free space, and ϵ_{r} is the relative permittivity (dielectric constant) of the solvent. The ion flux is defined as,

$$\mathbf{J}(\mathbf{x}, t) = -D_{\text{ion}} \nabla c - \frac{D_{\text{ion}} z F}{RT} \left(\nabla \varphi + \frac{\partial \mathbf{A}}{\partial t} \right) c + \mathbf{v} c, \tag{4.1c}$$

where R is the ideal gas constant, and T is the absolute temperature. The flux $\mathbf{J}$ represents possible modes of ion motion due to diffusion (Brownian motion) with the

ionic diffusivity of the solution, $D_{\text{ion}}(\mathbf{x}, t)$; and advection (drift) induced by the fluid velocity $\mathbf{v}(\mathbf{x}, t)$, electric current proportional to the potential gradient $\nabla\varphi(\mathbf{x}, t)$, and magnetic vector potential $\mathbf{A}(\mathbf{x}, t)$.

The PNP equations (4.1) are readily written either in the PDE format (2.1) or as the Itô process (3.1) (see [**?**] for details):

$$\mathcal{L} = \nabla^2, \qquad \lambda = -\frac{zF}{\epsilon_0 \epsilon_{\text{r}}}; \tag{4.2a}$$

$$\boldsymbol{\mu}(\mathbf{x}, t) = \mathbf{v} - \frac{D_{\text{ion}} zF}{RT}\left(\nabla\varphi + \frac{\partial \mathbf{A}}{\partial t}\right) + \nabla D_{\text{ion}}; \tag{4.2b}$$

$$\sigma_{ij} = \sqrt{2D_{ij}}, \qquad D_{ij}(\mathbf{x}, t) = \begin{cases} D_{\text{ion}} & i = j \\ 0 & \text{otherwise.} \end{cases} \tag{4.2c}$$

This formulation implicitly assumes that $p \equiv c$ or, more generally, that concentration is the (unnormalized) PDF of the probability of finding an ion in a given location at some time. This PDF reinterpretation of the chemical concentration aligns with the physical meaning of concentration as the density of particles at some location (e.g., [**?**, **?**]).

4.1. Benchmark solutions. We compare the results from our method with those computed via the open-source finite-volume PDE solver `FiPy` [**?**] on a fine mesh constructed via the open-source mesh generator `gmsh` [**?**]. The PNP system is solved through an iterative method that updates the electrostatic potential after every time step of the Nernst-Planck equation. Although such an iterative solver is not stable for highly nonlinear, strongly coupled problems, it is typically faster than robust direct methods. Since scalability is one of the core improvements of our method, a fast PDE solver was chosen for fair comparison and provides benchmark solutions to study error and convergence properties of our method. This comparison is facilitated by using the same time discretization in the `FiPy` solutions and in our integral kernel method unless otherwise specified.

Time step Δt and mesh resolution (number of finite volume nodes) used in each benchmark `FiPy` solution are detailed for each example. The underlying matrix solver is `SciPy`'s sparse LU solver. `SciPy` solvers [**?**] are recommended by the `FiPy` documentation for users of Python 3; all our numerical examples are run in Python 3.12.

The core step of the integral kernel method is the evaluation of the integrals in (3.14) and (3.15) through a quadrature/cubature rule. We opt for relatively simple rules with strong convergence properties. In the one- and two-dimensional examples reported below, we use respectively the Gauss-Legendre quadrature and the Gaussian quadrature [**?**] based on line integrals over distinct chords of the circle. In the three-dimensional example, we employ the Lebedev quadrature [**?**] coupled to Gauss-Legendre for the radial coordinate.

At every time step, the integral kernel solution is available at N_{pt} points from the set $S = \{\mathbf{x}_i\}_{i=1}^{N_{\text{pt}}}$. These points differ from the N_{fv} points (nodes) used in the finite volume (FV) solution. Hence, to quantify the discrepancy between the two solutions, we interpolate the FV solution from its grid to the points of the integral solution, yielding the benchmark solution $\{c_i^{\text{fv}}\}_{i=1}^{N_{\text{pt}}}$. The latter is compared with the

integral solution $\{c_i^{\rm pt}\}_{i=1}^{N_{\rm pt}}$, whose convergence is determined in terms of a relative mean squared error (MSE) $\mathcal{E}$,

$$\mathcal{E} = \frac{1}{N_{\rm pt}} \sum_{i=1}^{N_{\rm pt}} \left(\frac{c_i^{\rm fv} - c_i^{\rm pt}}{\overline{c}} \right)^2 . \tag{4.3}$$

The use of $\overline{c}$, the arithmetic mean of $\{c_i^{\rm pt}\}_{i=1}^{N_{\rm pt}}$, as a normalization constant facilitates both the interpretation of the MSE values and the comparison across the examples with vastly different initial concentrations [**?**].

4.2. One-dimensional example. Let $d = 1$ and $\Omega = [0, L]$ for some $L > 0$. The one-dimensional (1D) version of the PNP equations (4.1) is solved subject to initial condition

$$c(x, 0) = \begin{cases} 20 & x \in [L/4,\ 3L/4] \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

and boundary conditions

$$\varphi = 0 \quad \text{and} \quad J = 0, \quad \text{for} \quad x \in \{0, L\}. \tag{4.5}$$

For the simulations, we set $L = 4 \cdot 10^{-8}$ m, express the ion concentration $c(x, t)$ in mol/m^3, and discretize the time horizon $T = 5 \cdot 10^{-9}$ s with uniform time steps $\Delta t = 10^{-11}$ s.

No-flux conditions for the Fokker-Planck, $J = 0$, correspond to reflective boundary conditions for the underlying Itô process $X(t)$. This requires a construction of the Gaussian mixture transition density in (3.9) to match that behavior. Its 1D version for no-flux boundary conditions takes the form (see Appendix A for detail),

$$\hat{q}_{\rm 1D}^{\Delta t} = \phi(x; m^{\Delta t}, V^{\Delta t}) + \phi(x; -m^{\Delta t}, V^{\Delta t}) + \phi(x; 2L - m^{\Delta t}, V^{\Delta t}), \tag{4.6}$$

where the mean $m^{\Delta t}(x', t')$ and covariance $V^{\Delta t}(x', t')$ are defined in (3.7), with μ and D in (4.2). The first term in (4.6) represents the tentative update, while the second and third terms enforce, respectively, the reflective conditions on the left ($x = 0$) and right ($x = L$) ends of the domain Ω. These boundaries "reflect" back into Ω the density from the first term that would otherwise leak from the domain. The following theorem, proved in Appendix A, provides a means to determine the time step $\Delta t = t - t' > 0$ for any tolerance ϵ for the leaked (not reflected) concentration.

THEOREM 4.1. *Suppose the drift and diffusion coefficient of Itô process $X(t)$ are bounded, $|\mu(x', t')| \le \mu_\infty$ and $0 < D(x', t') \le D_\infty$, for every space-time point $(x', t') \in [0, L] \times (\mathbb{R}^+ \cup \{0\})$. Then, for any tolerance $\epsilon \in (0, 1)$, the selection of time step $\Delta t = t - t' \in (0, L/\mu_\infty)$ as a solution of*

$$\frac{L - \mu_\infty \Delta t}{\sqrt{2 D_\infty \Delta t}} = \Phi^{-1}\left(1 - \frac{\epsilon}{2}\right), \tag{4.7}$$

wherein $\Phi(\cdot)$ is the cumulative distribution function (CDF) of the standard normal, guarantees that the Gaussian mixture $\hat{q}_{\rm 1D}^{\Delta t}$ in (4.6) satisfies

$$\int_0^L \hat{q}_{\rm 1D}^{\Delta t}(x, t; x', t') \mathrm{d}x \ge 1 - \epsilon.$$

For this one-dimensional example, the Gaussian transition kernel K^ϕ is given by (see Appendix B.1),

$$K^\phi_{1\mathrm{D}}(x;\mathcal{M},\mathcal{V}) = \frac{x}{L}F_2(0,L;\mathcal{M},\mathcal{V}) - xF_1(x,L;\mathcal{M},\mathcal{V}) - F_2(0,x;\mathcal{M},\mathcal{V}), \tag{4.8a}$$

where

$$F_1(a,b;\cdot) = \int_a^b \phi(\tilde{x};\mathcal{M},\mathcal{V})\mathrm{d}\tilde{x} = \frac{1}{2}\,\mathrm{erf}\left(\frac{\mathcal{M}-a}{\sqrt{2\mathcal{V}}}\right) - \frac{1}{2}\,\mathrm{erf}\left(\frac{\mathcal{M}-b}{\sqrt{2\mathcal{V}}}\right) \tag{4.8b}$$

and

$$F_2(a,b;\cdot) = \int_a^b \tilde{x}\phi(\tilde{x};\mathcal{M},\mathcal{V})\mathrm{d}\tilde{x} = \mathcal{M}F_1(a,b;\cdot) + \mathcal{V}\phi(a;\cdot) - \mathcal{V}\phi(b;\cdot). \tag{4.8c}$$

According to Theorem 3.6 and Eq. (4.6), the approximate update kernel takes the form

$$\begin{aligned}\hat{K}^{\Delta t}_{1\mathrm{D}}(x;x',t') = \; & K^\phi_{1\mathrm{D}}(x;m^{\Delta t},V^{\Delta t}) \\ & + K^\phi_{1\mathrm{D}}(x;-m^{\Delta t},V^{\Delta t}) + K^\phi_{1\mathrm{D}}(x;2L-m^{\Delta t},V^{\Delta t}).\end{aligned} \tag{4.9}$$

This kernel satisfies not only the no-flux Nernst-Planck condition, achieved through the choice of an appropriate Gaussian mixture (4.6), but also the homogeneous Dirichlet Poisson condition since $K^\phi_{1\mathrm{D}}$ is constructed from the Green's function with homogeneous Dirichlet conditions (see Appendix B.1).

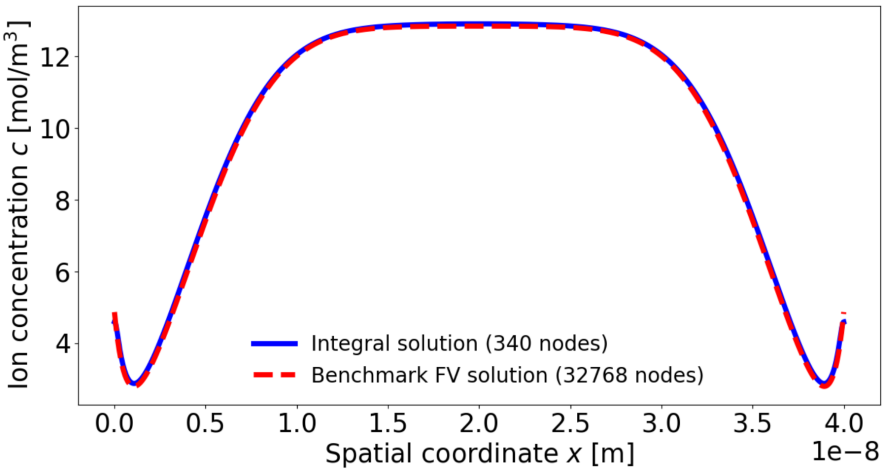

FIG. 1. *Ion concentration profiles $c(x,\cdot)$ predicted with the integral method and the benchmark finite volume solution at final time $T = 5\cdot 10^{-9}$ s. The no-flux (reflective) boundary conditions are properly enforced as seen by the accurate predictions of ion accumulation at the edges of the simulation domain.*

The one-dimensional example showcases our method's ability to handle numerical challenges such as discontinuous initial concentrations and ion accumulation near impermeable walls. Figure 1 demonstrates the visual agreement between the ion concentration profiles, $c(x, t = 5\cdot 10^{-9}\mathrm{s})$, computed via the integral kernel method with only $N_{\mathrm{pt}} = 340$ nodes and the benchmark finite-volume method with $N_{\mathrm{fv}} = 2^{15} = 32768$ cells, almost 100 times more.

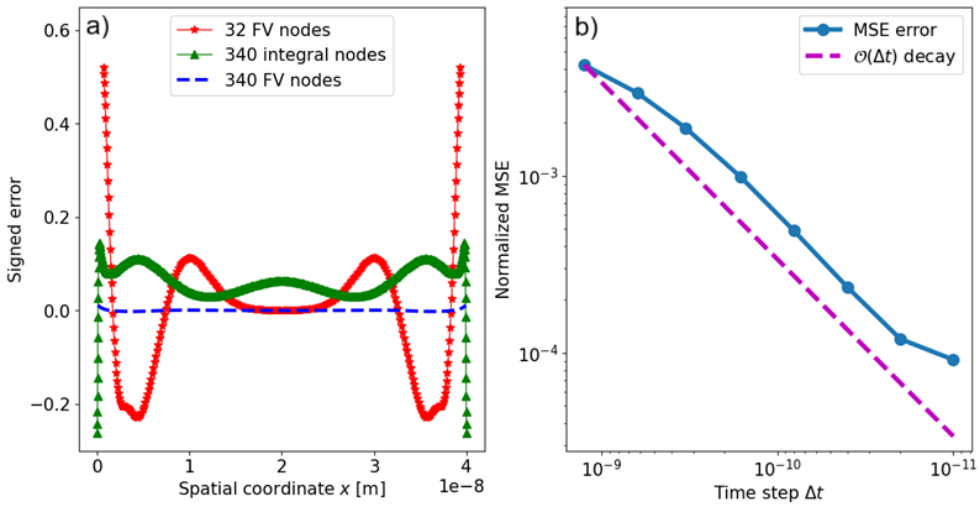

FIG. 2. *Numerical error in the one-dimensional example.* ***(a)*** *the signed error—numerical solution minus benchmark—of the solutions with* $\Delta t = 10^{-11}$ *s obtained with 32 finite volume nodes, 340 finite volume nodes, and 340 integral nodes, the last of which is also plotted in Figure 1.* ***(b)*** *the error of the 340 integrals nodes solution for different values of time step* Δt*, spanning two orders of magnitude, with the MSE decaying with convergence rate* $\mathcal{O}(\Delta t)$*.*

The green line with triangle markers in Figure 2a corresponds to the signed error—numerical solution minus benchmark—of the 340 integral nodes solution from Figure 1. As expected, the error is symmetrical and attains its highest magnitude at the boundaries due to the difficulty of modeling the ion accumulation accurately. This same behavior is observed in the error of the two FV solutions, which use 32 and 340 nodes: the error of the integral method with 340 nodes is closer to that of the FV with only 32 nodes. Indeed, the error of the 340 FV nodes solution (the blue dashed line) is significantly smaller than that of the integral method with same node count. However, it comes with the cost of meshing the domain and solving a potentially large linear system, with both tasks scaling very poorly with dimensionality and node count. Even though the FV method typically outperforms its competitors when the number of nodes is small, it scales poorly. This makes our approach much faster, for the same level of error, especially when accelerated by a FMM. We demonstrate this in detail in section 4.5.

We also analyzed the method's convergence with respect to time step Δt. Standard numerical integration methods for SDEs, such as the Euler-Maruyama method, are known to have absolute error decaying at the half-order convergence rate $\mathcal{O}(\Delta t^{1/2})$ in the presence of boundaries [**?**, **?**], which is $\mathcal{O}(\Delta t)$ when reported in terms of MSE as we do here.

In our one-dimensional example, the largest error is attained at the boundaries, which are harder for either method to model, so it is a prime study case to verify whether our method displays the same convergence order of SDE integrators. Using 340 integral nodes and varying only the time step Δt, we observe that the MSE does indeed very closely follow the $\mathcal{O}(\Delta t)$ decay (Fig. 2b), with the last data point flattening out the curve since the spatial error starts to dominate past that region. Typically, special treatment is needed to improve this convergence order [**?**, **?**], the

fact confirmed here.

In addition to these studies, Figure 3a shows our method's convergence with the number of nodes, $N_{\rm pt}$. The method is not convergent for a small number of integration nodes ($N_{\rm pt} < 240$, not shown here) but its normalized MSE, $\mathcal{E}$, decays fast once $N_{\rm pt}$ crosses this threshold, reaching $\mathcal{E} < 10^{-4}$ at $N_{\rm pt} = 340$. The solution with $N_{\rm pt} = 340$ is shown in Figure 1.

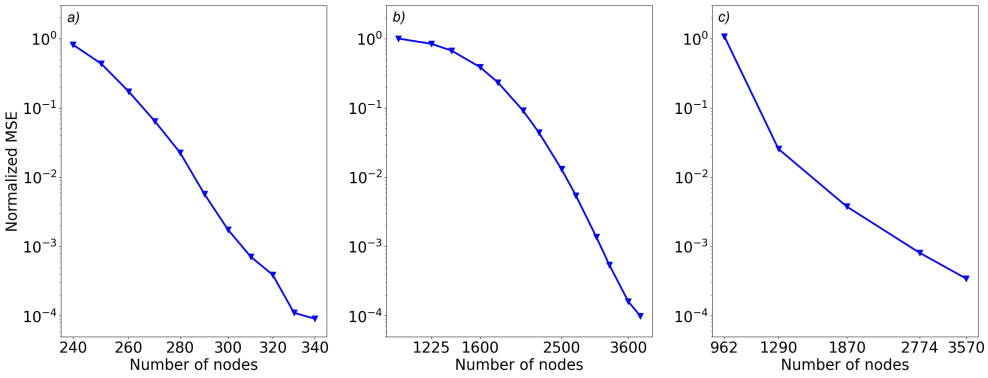

FIG. 3. *Normalized mean square error, $\mathcal{E}$, as function of the number of nodes, N_{pt}, used in our method to compute ion concentration c in (a) one-, (b) two-, and (c) three-dimensional examples.*

4.3. Two-dimensional example. Let $d = 2$ and $\Omega = \{\mathbf{x} = (x_1, x_2)^\top : x_1^2 + x_2^2 < R^2\}$ be a circle of radius R centered at the origin. The two-dimensional (2D) version of the PNP equations (4.1) is solved subject to boundary conditions

$$\varphi = 0 \quad \text{and} \quad p = 0, \quad \text{for} \quad \mathbf{x} \in \partial\Omega, \tag{4.10}$$

where $\partial\Omega = \{\mathbf{x} = (x_1, x_2)^\top : x_1^2 + x_2^2 = R^2\}$, and initial condition

$$c(\mathbf{x}, 0) = 50(c_1 + c_2), \tag{4.11a}$$

expressed as a mixture of two unnormalized Gaussians,

$$c_1(\mathbf{x}) = \exp\left[-\frac{5(x_1 + R/3)^2 + (x_2 - R/5)^2}{0.1R^2}\right] \tag{4.11b}$$

$$c_2(\mathbf{x}) = \exp\left[-\frac{x_1^2 + 5(x_2 + R/3)^2}{0.15R^2}\right]. \tag{4.11c}$$

In the simulations, we set $R = 2 \times 10^{-8}$ m, express the ion concentration $c(x, t)$ in mol/m^3, and discretize the time horizon $T = 2 \cdot 10^{-9}$ s with uniform time steps $\Delta t = 2 \cdot 10^{-10}$ s.

We show in Appendix B.2 that the Gaussian transition kernel $K^{\phi}_{\rm 2D}(\mathbf{x}; \boldsymbol{\mathcal{M}}, \boldsymbol{\mathcal{V}})$ for the 2D free-space PNP equations with diagonal covariance matrix $\boldsymbol{\mathcal{V}} = \sigma^2 \mathbf{I}_d$ can be conveniently parametrized in terms of the radial distance $r = \|\mathbf{x} - \boldsymbol{\mathcal{M}}\|_2$ rather than $\mathbf{x}$; specifically,

$$K^{\phi}_{\rm 2D}(r; \sigma^2) = \frac{1}{4\pi} \begin{cases} \mathrm{E}_1\left(\dfrac{r^2}{2\sigma^2}\right) + 2\ln(r) & r > 0 \\ -\gamma + \ln(2\sigma^2) & r = 0, \end{cases}$$

where $\|\cdot\|_2$ denotes the ℓ_2 vector norm, $\gamma \approx 0.58$ is the Euler-Mascheroni constant, and

$$\mathrm{E}_1(x) = \int_x^\infty \frac{\exp(-u)}{u}\mathrm{d}u \tag{4.12}$$

is the exponential integral function.

We use the method of images to modify the 2D free-space kernel K_{2D}^{ϕ} for the Poisson equation on the disk with homogeneous Dirichlet boundary conditions. Let us define an image $\mathbf{m}_{\mathrm{im}}^{\Delta t}(\mathbf{x}', t')$ of the mean $\mathbf{m}^{\Delta t}(\mathbf{x}', t')$ as[8]

$$\mathbf{m}_{\mathrm{im}}^{\Delta t} = \frac{R^2}{\|\mathbf{m}^{\Delta t}\|_2^2}\mathbf{m}^{\Delta t}. \tag{4.13}$$

Define

$$r = \|\mathbf{x} - \mathbf{m}^{\Delta t}\|_2 \quad \text{and} \quad r_{\mathrm{im}} = \frac{\|\mathbf{m}^{\Delta t}\|_2}{R}\|\mathbf{x} - \mathbf{m}_{\mathrm{im}}^{\Delta t}\|_2 \tag{4.14}$$

and let $(\cdot)$ denote the usual dot product, such that

$$\begin{aligned} r_{\mathrm{im}}^2 &= \frac{\|\mathbf{m}^{\Delta t}\|_2^2}{R^2}\left(\|\mathbf{x}\|_2^2 + \|\mathbf{m}_{\mathrm{im}}^{\Delta t}\|_2^2 - 2\mathbf{x}\cdot\mathbf{m}_{\mathrm{im}}^{\Delta t}\right) \\ &= \frac{\|\mathbf{m}^{\Delta t}\|_2^2}{R^2}\left(\|\mathbf{x}\|_2^2 + \frac{R^4}{\|\mathbf{m}^{\Delta t}\|_2^4}\|\mathbf{m}^{\Delta t}\|_2^2 - 2\frac{R^2}{\|\mathbf{m}^{\Delta t}\|_2^2}\mathbf{x}\cdot\mathbf{m}^{\Delta t}\right) \\ &= \frac{\|\mathbf{m}^{\Delta t}\|_2^2}{R^2}\|\mathbf{x}\|_2^2 + R^2 - 2\mathbf{x}\cdot\mathbf{m}^{\Delta t}. \end{aligned}$$

If $\mathbf{x} \in \partial\Omega$, then $\|\mathbf{x}\|_2 = R$ and

$$\begin{aligned} r_{\mathrm{im}}^2 &= \frac{\|\mathbf{m}^{\Delta t}\|_2^2}{R^2}R^2 + \|\mathbf{x}\|_2^2 - 2\mathbf{x}\cdot\mathbf{m}^{\Delta t} \\ &= \|\mathbf{x}\|_2^2 + \|\mathbf{m}^{\Delta t}\|_2^2 - 2\mathbf{x}\cdot\mathbf{m}^{\Delta t} \\ &= r^2. \end{aligned}$$

This observation enables us to approximate the update kernel as

$$\hat{K}_{\mathrm{2D}}^{\Delta t}(\mathbf{x}; \mathbf{x}', t') = K_{\mathrm{2D}}^{\phi}(r, \sigma^2) - K_{\mathrm{2D}}^{\phi}(r_{\mathrm{im}}, \sigma^2), \tag{4.15}$$

such that $\hat{K}_{\mathrm{2D}}^{\Delta t}(\mathbf{x}; \mathbf{x}', t') = 0$ by construction on the circle's circumference. This immediately satisfies $\varphi(\mathbf{x} \in \partial\Omega, t) = 0$, given the update step from Eq. (3.13).

On the other hand, the homogeneous concentration condition ("absorbing" boundary) is satisfied by not reinserting the "leaked" density into the system. No further modification of the kernel is needed.

Figure 4 provides a visual comparison of the spatial maps of the ion concentration $c(\mathbf{x}, t)$ and the electric potential $\varphi(\mathbf{x}, t)$, alternatively computed, at final time $T = 2 \cdot 10^{-9}$ s, via the integral kernel method with $N_{\mathrm{pt}} = 3600$ nodes and the benchmark finite-volume solution with $N_{\mathrm{fv}} = 46886$ cells. To illustrate the locations of the 3600 nodes used in the integral kernel method, the values of $c(\mathbf{x}, t)$ and $\varphi(\mathbf{x}, t)$ at these points are shown as colored dots, without interpolation. Apart from this pixelation, the two solutions are similar, validating the correctness of our method. A more quantitative depiction of our method's accuracy is provided in Fig. 3b, which shows the decay of normalized MSE $\mathcal{E}$ with the number of integration nodes N_{pt}.

[8]This result is in direct analogy with the construction of an image charge position used to derive the Green's function for the Laplacian on a circle [**?**, Sec. 9.5.9].

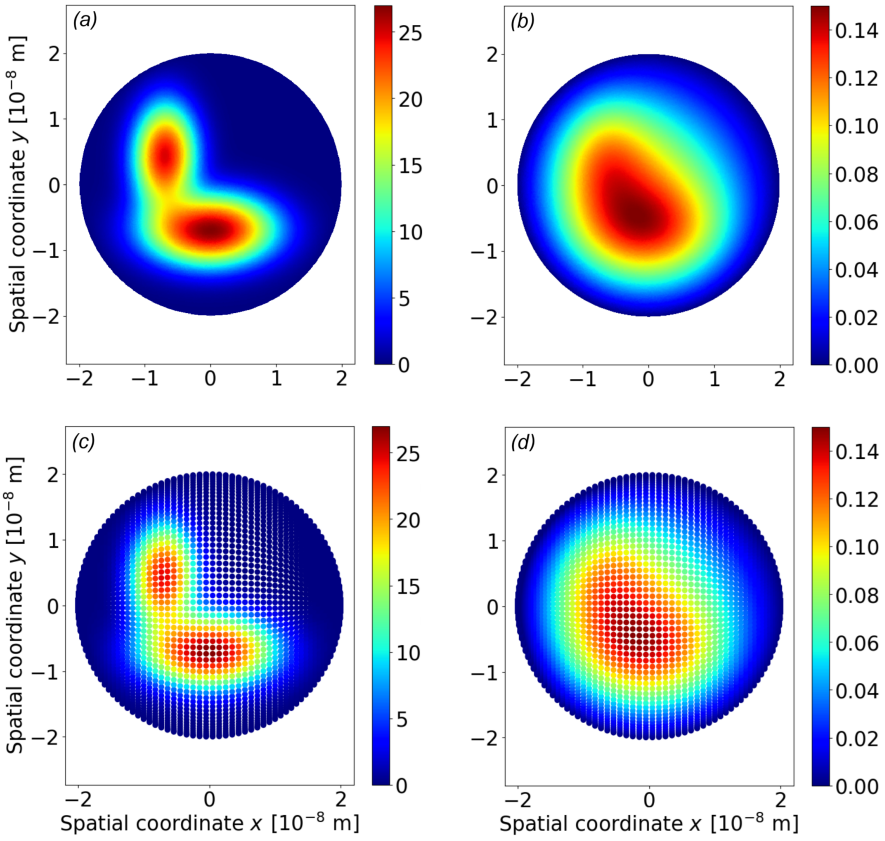

FIG. 4. *Spatial distributions, at final time $T = 2 \cdot 10^{-9}$ s, of the ion concentration $c(\mathbf{x}, t)$ (left column) and the electric potential $\varphi(\mathbf{x}, t)$ (right column), alternatively computed via (top row) the benchmark finite-volume solution with $N_{fv} = 46886$ cells and (bottom row) the integral kernel method with $N_{pt} = 3600$ nodes.*

4.4. Three-dimensional example. Let $d = 3$ and $\Omega = \{\mathbf{x} = (x_1, x_2, x_3)^\top : x_1^2 + x_2^2 + x_3^2 < R^2\}$ be a sphere of radius R centered at the origin. The three-dimensional (3D) version of the PNP equations (4.1) is solved subject to initial condition

$$c(\mathbf{x}, 0) = \begin{cases} 0.1 & \mathbf{x} \in [-R/3,\ R/3]^3 \\ 0 & \text{otherwise} \end{cases} \tag{4.16}$$

and homogeneous Dirichlet boundary conditions

$$\varphi = 0 \quad \text{and} \quad p = 0, \quad \text{for} \quad \mathbf{x} \in \partial\Omega, \tag{4.17}$$

where $\partial\Omega = \{\mathbf{x} = (x_1, x_2, x_3)^\top : x_1^2 + x_2^2 + x_3^2 = R^2\}$. In the simulations, we set $R = 10^{-7}$ m and discretize the time horizon $T = 4 \cdot 10^{-7}$ s with uniform time steps $\Delta t = 5 \cdot 10^{-8}$ s.

We show in Appendix B.3 that the Gaussian transition kernel $K_{3\mathrm{D}}^{\phi}(\mathbf{x}; \boldsymbol{\mathcal{M}}, \boldsymbol{\mathcal{V}})$ for the 3D free-space PNP equations with diagonal covariance matrix $\boldsymbol{\mathcal{V}} = \sigma^2 \mathbf{I}_d$ can also be conveniently parametrized in terms of the radial distance $r = \|\mathbf{x} - \boldsymbol{\mathcal{M}}\|_2$ rather

than $\mathbf{x}$; specifically,

$$K^{\phi}_{3\mathrm{D}}(r;\sigma^2) = \begin{cases} -\dfrac{1}{4\pi r}\,\mathrm{erf}\left(\dfrac{r}{\sqrt{2\sigma^2}}\right) & r > 0 \\ -\dfrac{1}{(2\pi)^{3/2}\sigma} & r = 0. \end{cases}$$

The argument used in the previous example follows,

$$\hat{K}^{\Delta t}_{3\mathrm{D}}(\mathbf{x};\mathbf{x}',t') = K^{\phi}_{3\mathrm{D}}(r,\sigma^2) - K^{\phi}_{3\mathrm{D}}(r_{\mathrm{im}},\sigma^2). \tag{4.18}$$

As before, $\hat{K}^{\Delta t}_{3\mathrm{D}}(\mathbf{x};\mathbf{x}',t') = 0$ for $\mathbf{x} \in \partial\Omega$, so that $\varphi(\mathbf{x} \in \partial\Omega, t) = 0$, and the homogeneous concentration conditions are satisfied by letting the concentration flow out of the domain without reinsertion.

The integral kernel method with $\hat{K}^{\Delta t}_{3\mathrm{D}}$ from (4.18) yields a solution for ion concentration $c(\mathbf{x},t)$, whose error relative to the benchmark FV solution with 66076 cells, is shown in Fig. 3c. Similar to the 1D and 2D examples, the method requires a certain number of nodes to converge ($N_{\mathrm{pt}} \geq 962$), fast once the number of nodes is sufficiently high. The convergence is bounded by the quality of the benchmark solution, and as such it does not achieve the same very low level of error as the other cases, since a 3D problem with discontinuous conditions makes it challenging for the finite volume solver to achieve a very accurate solution, even with a high number of cells.

4.5. FMM-accelerated 2D example. In addition to illustrating the accuracy and convergence of the proposed algorithm, we demonstrate the algorithm's acceleration via FMM (section 3.5). In this numerical experiment, we utilize the two-dimensional kernel-independent FMM implementation in the external `C++` library `BBFMM2D` [**?**].

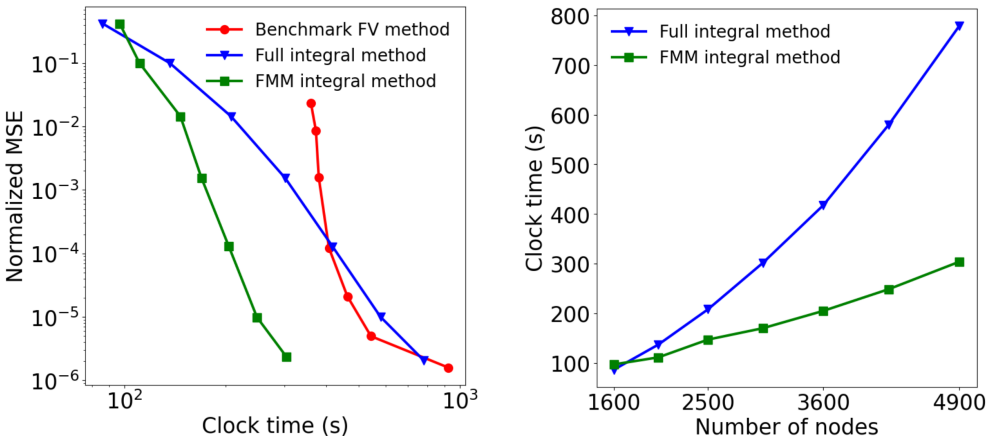

FIG. 5. *Left: The clock time taken by each method to reach certain levels of error, $\mathcal{E}$. Right: the clock time as function of the number of integration nodes for both the integral kernel method and its FMM-accelerated version.*

We use the above-mentioned two-dimensional example with the same time step of $\Delta t = 2 \times 10^{-10}$ s and same number of nodes ($N_{\mathrm{pt}} = 1600$, 2025, 2500, 3025, 3600, 4225, 4900) for both the full and FMM integral methods. The benchmark FV solution is now computed with $\Delta t = 10^{-12}$ s with everything else equal to produce

an exceptionally accurate solution. Since FMM introduces (small) errors [**?**, **?**], we compare the original and the FMM methods to a high-fidelity benchmark solution generated on a much finer time grid to precisely measure the accuracy penalty of using FMM.

Figure 5 shows that despite very different clock times at the same number of integration nodes (right panel), the integral kernel method and its FMM-accelerated version are almost indistinguishable in their convergence (left panel), making a strong case for FMM acceleration. The numerically measured time complexity is close to linear in the number of integration nodes (line with squares in right panel of Fig. 5). Its scalability and order of convergence are competitive with those of the conventional FV solver, especially for high accuracy (left panel of Fig 5). Although exact clock times depend on hardware, time discretization, and number of integration nodes (or mesh in the FV case), the desirable scalability and complexity are inherent properties of the algorithm.

5. Discussion. The numerical results reported in Fig. 3 highlight the existence of a threshold number of nodes ($N_{\text{pt}} = 240, 1225$, and 962 in the 1D, 2D and 3D examples, respectively), below which the method does not reach convergence[9] and above which it converges fast. The nonstandard character and multiple components of our algorithm, such as different integration techniques, complicate any attempt at a comprehensive error analysis. Instead, we found a rule of thumb to determine when the algorithm is expected to converge to a reasonable, physical solution. The standard deviation σ of the Gaussian kernel K^{ϕ} in Eq. (3.11) must be of similar magnitude or bigger than the largest distance between "neighboring" integration nodes. The condition makes intuitive sense, as σ is akin to a characteristic length of the transition PDF of the underlying process; hence, an overly large spacing between the nodes precludes an accurate representation of the transition densities.

Our direct and fully explicit method for solving nonlinear coupled PDEs is robust and exhibits linear time and storage complexity. The former is achieved through FMM, while the latter is a consequence of being matrix-free and quadrature-based. The method is embarrassingly parallel in its quadratic formulation but also easily parallelizable with FMM in three dimensions thanks to off-the-shelf, high-performance, kernel-independent FMM implementations like `PBBFMM3D` [**?**] and `ExaFMM` [**?**]. The availability of optimized FMM packages makes implementation straightforward, since the core routine consists of evaluating Eqs. (3.14) and (3.15) at each time step.

The global information update property of these core equations makes the method ideal for ill-conditioned, convection-dominated problems. In the examples considered, our algorithm remains numerically stable (albeit less accurate) for time steps as large as the total simulation time, even for fine quadrature grids, with no sign of a Courant–Friedrichs–Lewy-like condition [**?**] that would limit Δt. For the one-dimensional case, we proved (see Theorem 4.1) that a suitable step size depends only on bounds on the characteristics of the underlying stochastic process, regardless of spatial discretization.

While this result does not directly provide a sufficient criterion for stability, it guides the choice of time step based on the maximum possible density loss at each step, which is a major source of numerical error for large time steps. Other than this theoretical result, we did not find any empirical limit on Δt for stability, even on the two examples with discontinuities. We did observe a progressive loss of accuracy at

[9] The numerical solutions exhibit either a blow-up or a flat (zero) solution everywhere, neither of which is meaningful.

larger steps, as expected. The only stability consideration that we empirically found is the aforementioned relationship of σ and node spacing.

When applicable, our method outperforms conventional integral methods for elliptic equations, e.g., Eq. (2.1a), not only because it solves the full system (2.1) but also due to the absence of singularities in the integral kernel (3.12). Since the kernel is a convolution of the Green's function with a Gaussian PDF, it does not encounter the numerical difficulties faced by FMM-accelerated integral methods based on singular kernels arising from layer potentials or similar Green's functions-based formulations [**?**, **?**, **?**]. However, other integral methods can still be applied to compute the solution $\varphi_{\mathrm{b}}(\mathbf{x})$ of the homogeneous version of (2.1a), as defined in Theorem 3.3, since our proposed method is designed to solve only the inhomogeneous part.

Our results are sensitive to the choice of a numerical integration method. The three quadrature rules used exactly integrate all polynomials up to a certain degree in their respective geometries, and they are known to perform well in practice. We explored other numerical integration techniques and found them deficient. For instance, conventional Monte Carlo integration requires several orders of magnitude more nodes to yield comparable errors in the PDE solution, and simple one-dimensional rules like Simpson's or trapezoidal require approximately one order of magnitude more points for similar results. In addition to the three efficient quadratures considered here, others might be used without any performance loss. For example, for the two-dimensional case, a Zernike quadrature [**?**] yields the same if not slightly better performance.

Despite its strengths, the current implementation of our method is limited to canonical (including infinite) domains. This limitation stems from the need to match boundary conditions (2.1d) by directly constructing either the mixture PDF (3.9), as done the 1D example, or the approximate update kernel (3.12), as done in the 2D and 3D examples. This limitation is expected to gradually decrease in importance as machine learning methods become capable of computing Green's functions in complex geometries [**?**, **?**], a corner step into approximating the update kernel (3.12). The ability to handle complex geometries is one possible direction for future research.

Another direction for future work is to extend our method to a larger class of boundary conditions. For no-flux boundaries, the one-dimensional description is simple, with the transition PDF constructed as a three-component Gaussian mixture. The same technique would not immediately apply in higher dimensions, because of the difficulty in deriving a transition PDF that correctly models the reflected diffusion on a high-dimensional surface. This task is notoriously challenging in nontrivial geometries; problems of this nature are referred to as Skorokhod problems. Inhomogeneous Dirichlet or Neumann conditions can also be challenging as they require the flux at the boundary to be known so that the densities are correctly modeled. MCell [**?**], a generalized Monte Carlo software for simulation of diffusion and chemical reactions in crowded environments, has provided useful solutions that might help extend the current work to a wider set of boundary conditions.

Other directions include i) the method's expansion to nonlinear systems beyond beyond those described by Eq. (2.1) and ii) the incorporation of problem-specific, structure-preserving features deemed essential for a physical solution in challenging applications. These properties are mass/energy conservation, positivity, causality, minimum/maximum principles, etc.

6. Conclusion. We introduced a novel integral kernel numerical method for a class of nonlinear parabolic-elliptic PDE systems that arise in electrodiffusion, chemotaxis, self-gravitating particles, Brownian vortices, among other applications. Our

one-, two-, and three-dimensional examples demonstrate the method's robustness in handling discontinuous initial conditions, as well as its accuracy and competitiveness with grid-based PDE solvers thanks to FMM acceleration.

Being application-agnostic, the method does not intend to replicate the problem-specific benefits of specialized, complex methods. Instead, it offers a general, fast, low-storage, parallelizable algorithm of simple implementation for any system in the form of Eq. (2.1). It provides a new and efficient way to approach problems of this kind, and we hope to spur further development of other fast algorithms for nonlinear PDE systems.

Appendix A. Transition density for 1D PNP equations.

Let

$$\epsilon' = \int_{-\infty}^{0} \phi(x;\mathcal{M},\mathcal{V})\mathrm{d}x - \int_{\Omega} \phi(x;-\mathcal{M},\mathcal{V})\mathrm{d}x \tag{A.1a}$$

$$\epsilon'' = \int_{L}^{\infty} \phi(x;\mathcal{M},\mathcal{V})\mathrm{d}x - \int_{\Omega} \phi(x;2L-\mathcal{M},\mathcal{V})\mathrm{d}x \tag{A.1b}$$

denote the density fractions of the tentative Gaussian update PDF $\phi(x;\mathcal{M},\mathcal{V})$ that are respectively supported on $x<0$ and $x>L$ (first integrals), but are not reflected back into the domain $\Omega=[0,L]$ (second integrals) by the transition density $\hat{q}_{1\mathrm{D}}^{\Delta t}$ (4.6). The density fractions supported on $x>2L$ and $x<-L$ cannot be reflected back into Ω via an equidistant reflection across the boundary, because the reflected points would end up outside of Ω on the opposite side. Mathematically,

$$\int_{\Omega} \phi(x;-\mathcal{M},\mathcal{V})\mathrm{d}x = \int_{-L}^{0} \phi(x;\mathcal{M},\mathcal{V})\mathrm{d}x \tag{A.1c}$$

and

$$\int_{\Omega} \phi(x;2L-\mathcal{M},\mathcal{V})\mathrm{d}x = \int_{L}^{2L} \phi(x;\mathcal{M},\mathcal{V})\mathrm{d}x, \tag{A.1d}$$

such that

$$\epsilon' = \int_{-\infty}^{-L} \phi(x;\mathcal{M},\mathcal{V})\mathrm{d}x, \qquad \epsilon'' = \int_{2L}^{\infty} \phi(x;\mathcal{M},\mathcal{V})\mathrm{d}x. \tag{A.1e}$$

We would like the support of $\hat{q}_{1\mathrm{D}}^{\Delta t}$ to be contained in Ω as much as possible to enforce the reflective (no-flux) conditions with minimum density leak, so the nonnegative ϵ' and ϵ'' should be small since

$$\int_{\Omega} \hat{q}_{1\mathrm{D}}^{\Delta t}(x,t;x',t')\mathrm{d}x = 1-\epsilon'-\epsilon''. \tag{A.1f}$$

Under some conditions, this density loss can be made arbitrarily small with a proper choice of Δt, as shown in Theorem 4.1, whose proof is presented below.

Proof of Theorem 4.1. For given mean $m^{\Delta t}(x',t')$ and variance $V^{\Delta t}(x',t')$ of the Gaussian PDF $\phi(x;\cdot)$, let us choose $\Delta t \in (0, L/\mu_\infty)$ as a solution of the transcendental equation

$$\frac{L-\mu_\infty \Delta t}{\sqrt{2D_\infty \Delta t}} = \Phi^{-1}\left(1-\frac{\epsilon}{2}\right), \tag{A.2}$$

where Φ is the standard Gaussian CDF and ϵ is a (small) positive number. To prove the theorem, it suffices to show that such Δt results in

$$\int_{-\infty}^{-L} \phi(x; m^{\Delta t}, V^{\Delta t})\mathrm{d}x + \int_{2L}^{\infty} \phi(x; m^{\Delta t}, V^{\Delta t})\mathrm{d}x \le \epsilon. \tag{A.3}$$

Consider a particle within the simulation domain, $x' \in \Omega$. According to (3.7), the mean and variance of its next *tentative* update are $m^{\Delta t} = x' + \mu(x', t')\Delta t$ and $V^{\Delta t} = 2D(x', t')\Delta t$. Since $|\mu(x', t')| \le \mu_\infty$ and $\Delta t < L/\mu_\infty$, $m^{\Delta t} \in (-L, 2L)$.

Since $m^{\Delta t} > -L$, the first integral in (A.3),

$$\int_{-\infty}^{-L} \phi(x; m^{\Delta t}, V^{\Delta t})\mathrm{d}x = \int_{-\infty}^{-L} \phi(x; x' + \mu(x', t')\Delta t, 2D(x', t')\Delta t)\mathrm{d}x,$$

is bounded from above by the case of the lowest attainable mean, $m^{\Delta t} = 0 - \mu_\infty \Delta t$, i.e., by

$$\int_{-\infty}^{-L} \phi(x; m^{\Delta t}, V^{\Delta t})\mathrm{d}x \le \int_{-\infty}^{-L} \phi(x; -\mu_\infty \Delta t, 2D(x', t')\Delta t)\mathrm{d}x. \tag{A.4}$$

Another consequence of $m^{\Delta t} > -L$ is that the value of the integral monotonically increases with $V^{\Delta t}$. Since $D(x', t') \le D_\infty$,

$$\int_{-\infty}^{-L} \phi(x; m^{\Delta t}, V^{\Delta t})\mathrm{d}x \le \int_{-\infty}^{-L} \phi(x; -\mu_\infty \Delta t, 2D_\infty \Delta t)\mathrm{d}x. \tag{A.5}$$

Next, we use the property of the Gaussian distribution, $\phi(x; \mu, \sigma^2) = \phi((x-\mu)/\sigma; 0, 1)$, to transform the density into a standard normal,

$$\begin{aligned} \int_{-\infty}^{-L} \phi(x; m^{\Delta t}, V^{\Delta t})\mathrm{d}x &\le \int_{-\infty}^{-L} \phi\left(\frac{x + \mu_\infty \Delta t}{\sqrt{2D_\infty \Delta t}}; 0, 1\right)\mathrm{d}x \\ &= \Phi\left(\frac{-L + \mu_\infty \Delta t}{\sqrt{2D_\infty \Delta t}}\right). \end{aligned} \tag{A.6}$$

It follows from the symmetry of the Gaussian distribution that $\Phi(a) + \Phi(-a) = 1$, for any real a. Hence, from (A.2),

$$\Phi\left(\frac{-L + \mu_\infty \Delta t}{\sqrt{2D_\infty \Delta t}}\right) = 1 - \Phi\left(\frac{L - \mu_\infty \Delta t}{\sqrt{2D_\infty \Delta t}}\right) = 1 - \left(1 - \frac{\epsilon}{2}\right) = \frac{\epsilon}{2},$$

so that

$$\int_{-\infty}^{-L} \phi(x; m^{\Delta t}, V^{\Delta t})\mathrm{d}x \le \frac{\epsilon}{2}.$$

For the second integral in (A.3), we note that $m^{\Delta t} < 2L$ and follow the analogous

steps to obtain

$$\begin{aligned}\int_{2L}^{\infty}\phi(x;m^{\Delta t},V^{\Delta t})\mathrm{d}x &= \int_{2L}^{\infty}\phi(x;x'+\mu(x',t')\Delta t,2D(x',t')\Delta t)\mathrm{d}x\\ &\le \int_{2L}^{\infty}\phi(x;L+\mu_\infty\Delta t,2D_\infty\Delta t)\mathrm{d}x\\ &\le \int_{2L}^{\infty}\phi\left(\frac{x-L-\mu_\infty\Delta t}{\sqrt{2D_\infty\Delta t}};0,1\right)\mathrm{d}x\\ &\le 1-\Phi\left(\frac{L-\mu_\infty\Delta t}{\sqrt{2D_\infty\Delta t}}\right)\\ &\le \frac{\epsilon}{2},\end{aligned}$$

which completes the proof. ∎

Appendix B. Derivation of Gaussian transition kernels.

We use the one-, two-, and three-dimensional examples to showcase alternative strategies for the derivation of Gaussian kernels K^ϕ.

B.1. One-dimensional transition kernel. The Green's function for the 1D Poisson equation in section (4.2) is

$$G(x,\tilde{x})=\begin{cases}\dfrac{x-L}{L}\tilde{x} & x>\tilde{x}\\ \dfrac{\tilde{x}-L}{L}x & x\le\tilde{x}.\end{cases}$$

Substitution of this expression into the 1D version of (3.11) yields the Gaussian transition kernel,

$$\begin{aligned}K^{\phi}_{1\mathrm{D}}(x;\mathcal{M},\mathcal{V}) &= \int_0^x\frac{x-L}{L}\tilde{x}\phi(\tilde{x};\mathcal{M},\mathcal{V})\mathrm{d}\tilde{x}+\int_x^L\frac{\tilde{x}-L}{L}x\phi(\tilde{x};\mathcal{M},\mathcal{V})\mathrm{d}\tilde{x}\\ &= \frac{x-L}{L}F_2(0,x;\cdot)+\frac{x}{L}F_2(x,L;\cdot)-xF_1(x,L;\cdot)\\ &= \frac{x}{L}F_2(0,L;\cdot)-xF_1(x,L;\cdot)-F_2(0,x;\cdot),\end{aligned}\tag{B.1}$$

where the functions $F_1(a,b;\mathcal{M},\mathcal{V})$ and $F_2(a,b;\mathcal{M},\mathcal{V})$ are defined in Eq. (4.8).

B.2. Two-dimensional transition kernel. An alternative way to compute a Gaussian transition kernel is to rewrite Eq. (3.11) as $\mathcal{L}K^\phi(\mathbf{x};\boldsymbol{\mathcal{M}},\boldsymbol{\mathcal{V}})=\phi(\mathbf{x};\boldsymbol{\mathcal{M}},\boldsymbol{\mathcal{V}})$. This indicates that the Gaussian transition kernel is a solution to the linear differential operator under a Gaussian forcing function. Specifically, the solution in $\mathbb{R}^2$ to the 2D Poisson equation $\nabla^2\varphi=A\exp(-cr^2)$, with $A\in\mathbb{R}$ and $c\in\mathbb{R}^+$, is [**?**]

$$\varphi(r)=\frac{A}{4c}\begin{cases}\mathrm{E}_1(cr^2)+2\ln r & r>0\\ -\gamma-\ln c & r=0,\end{cases}\tag{B.2}$$

where γ is the Euler-Mascheroni constant and $\mathrm{E}_1(\cdot)$ the exponential integral function, as defined in Eq. (4.12). In our case, $A\exp(-cr^2)$ is the d-variate ($d=2$) Gaussian PDF $\phi(\mathbf{x};\boldsymbol{\mathcal{M}},\boldsymbol{\mathcal{V}})$ with diagonal covariance matrix $\boldsymbol{\mathcal{V}}=\sigma^2\mathbf{I}_d$,

$$\phi(\mathbf{x};\boldsymbol{\mathcal{M}},\sigma^2\mathbf{I}_d)=\frac{1}{(2\pi\sigma^2)^{d/2}}\exp\left(\frac{||\mathbf{x}-\boldsymbol{\mathcal{M}}||_2^2}{2\sigma^2}\right).\tag{B.3}$$

Hence, setting $A = 1/(2\pi\sigma^2)$, $c = 1/(2\sigma^2)$, and $r = ||\mathbf{x} - \boldsymbol{\mathcal{M}}||_2$ in (B.2), we obtain the Gaussian transition kernel $K^{\phi}_{\mathrm{II}}(\mathbf{x}; \boldsymbol{\mathcal{M}}, \boldsymbol{\mathcal{V}})$ on 2D free space,

$$K^{\phi}_{\mathrm{2D}}(r;\sigma^2) = \frac{1}{4\pi}\begin{cases} \mathrm{E}_1\left(\dfrac{r^2}{2\sigma^2}\right) + 2\ln r & r > 0 \\ -\gamma + \ln(2\sigma^2) & r = 0. \end{cases} \tag{B.4}$$

For any $r > 0$,

$$\lim_{\sigma\to 0^+} K^{\phi}_{\mathrm{2D}}(r;\sigma^2) = \frac{1}{2\pi}\ln r$$

which is the Green's function for the 2D free-space Laplacian. The degeneration of the transition kernel into the Green's function in the zero uncertainty limit is an immediate consequence of Theorem 3.3.

B.3. Three-dimensional transition kernel. In line with the derivation in B.2, we rely on the solution in $\mathbb{R}^3$ to the 3D Poisson equation $\nabla^2\varphi = A\exp(-cr^2)$ with $A \in \mathbb{R}$ and $c \in \mathbb{R}^+$ [**?**],

$$\varphi(r) = \begin{cases} -\dfrac{A\sqrt{\pi}}{4c^{3/2}r}\,\mathrm{erf}\,(\sqrt{c}r) & r > 0 \\ -\dfrac{A}{2c} & r = 0. \end{cases} \tag{B.5}$$

The function $A\exp(-cr^2)$ coincides with the d-variate ($d = 3$) Gaussian PDF in (B.3) if $A = (2\pi\sigma^2)^{-3/2}$, $c = 1/(2\sigma^2)$, and $r = ||\mathbf{x} - \boldsymbol{\mathcal{M}}||_2$. Therefore, the Gaussian transition kernel $K^{\phi}_{\mathrm{3D}}(\mathbf{x}; \boldsymbol{\mathcal{M}}, \boldsymbol{\mathcal{V}})$ on 3D free space is

$$K^{\phi}_{\mathrm{3D}}(r;\sigma^2) = \begin{cases} -\dfrac{1}{4\pi r}\,\mathrm{erf}\left(\dfrac{r}{\sqrt{2\sigma^2}}\right) & r > 0 \\ -\dfrac{1}{(2\pi)^{3/2}\sigma} & r = 0. \end{cases} \tag{B.6}$$

For any $r > 0$,

$$\lim_{\sigma\to 0^+} K^{\phi}_{\mathrm{3D}}(r;\sigma^2) = -\frac{1}{4\pi r},$$

which is the Green's function for the three-dimensional free-space Laplacian.