Journal of Computational Physics 334 (2017) 68–80

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

On the use of reverse Brownian motion to accelerate hybrid simulations

Joseph Bakarji, Daniel M. Tartakovsky [*]

Department of Energy Resources Engineering, Stanford University, 367 Panama Street, Stanford, CA 94305, USA

ARTICLE INFO

Article history:
Received 19 May 2016
Received in revised form 16 December 2016
Accepted 18 December 2016
Available online 30 December 2016

Keywords:
Hybrid simulation
Local Monte Carlo
Reverse Brownian motion

ABSTRACT

Multiscale and multiphysics simulations are two rapidly developing fields of scientific computing. Efficient coupling of continuum (deterministic or stochastic) constitutive solvers with their discrete (stochastic, particle-based) counterparts is a common challenge in both kinds of simulations. We focus on interfacial, tightly coupled simulations of diffusion that combine continuum and particle-based solvers. The latter employs the reverse Brownian motion (rBm), a Monte Carlo approach that allows one to enforce inhomogeneous Dirichlet, Neumann, or Robin boundary conditions and is trivially parallelizable. We discuss numerical approaches for improving the accuracy of rBm in the presence of inhomogeneous Neumann boundary conditions and alternative strategies for coupling the rBm solver with its continuum counterpart. Numerical experiments are used to investigate the convergence, stability, and computational efficiency of the proposed hybrid algorithm.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

Multiscale and multiphysics simulations are two rapidly developing fields of scientific computing. We use the terms “multiphysics” and “multiscale” as distinct classifiers. The former involves two or more processes (e.g., fluid flow and solid deformations), while the latter employs two or more descriptions of the same process (e.g., kinetic Monte Carlo and Navier–Stokes equations used to describe fluid flow). Efficient coupling of continuum (deterministic or stochastic) constitutive solvers with their discrete (stochastic, particle-based) counterparts is a common challenge in both kinds of simulations (e.g., [13, Sec. 3.3]).

Such a coupling can be either distributed, when constitutive solvers are defined on overlapping computational (sub)domains (e.g., [5,19]), or interfacial, i.e., confined to the boundaries Γ between non-overlapping computational subdomains (e.g., [8,18]). We focus on interfacial, tightly coupled simulations that combine a continuum deterministic solver and a discrete stochastic solver. While the approach proposed here is also applicable to multiphysics simulations of this kind, we pose the problem in terms relevant to hybrid algorithms. These represent a subset of multiscale simulations in which a continuum description of a certain phenomenon breaks down in a (small) part of the computational domain. That occurs, for example, in simulations of dilute gas flows, which are adequately described by (continuum) Navier–Stokes equations everywhere except in the vicinity of solid walls where a (discrete) direct simulation Monte Carlo method has to be used [9,19].

Constitutive multiscale solvers of a typical hybrid algorithm are coupled by enforcing continuity of relevant state variables across the interface Γ. To be concrete, let us consider a hybrid simulation that combines a reaction–diffusion equation in

[*] Corresponding author.
E-mail address: tartakovsky@stanford.edu (D.M. Tartakovsky).

http://dx.doi.org/10.1016/j.jcp.2016.12.032
0021-9991/© 2016 Elsevier Inc. All rights reserved.

one subdomain, D_1, with its discrete counterpart (e.g., Brownian motion and stochastic simulation algorithm describing, respectively, diffusion and reactions [5]) in the other, D_2 ($D_1 \cap D_2 = \emptyset$ and $D_1 \cup D_2 = D$ where D is the simulation domain). At every time step, an initial guess of the system state's value along the interface Γ between the subdomains D_1 and D_2, $u(\mathbf{x}, t) = u_\Gamma(\mathbf{x}, t)$ for $\mathbf{x} \in \Gamma$, would result in a system of two boundary-value problems, which can be solved independently from each other. The outcome is solutions $u_1(\mathbf{x}, t)$ for $\mathbf{x} \in D_1$ and $u_2(\mathbf{x}, t)$ for $\mathbf{x} \in D_2$ that are, by construction, continuous at $\mathbf{x} \in \Gamma$ but might violate continuity of the normal components of Fickian flux, $q_n \sim \mathbf{n} \cdot \nabla u$ where $\mathbf{n}(\mathbf{x})$ is the unit vector normal to Γ. The latter is enforced by modifying the initial guess u_Γ and iterating until both the state variable u and its flux q_n are continuous within a prescribed tolerance.

This tight-coupling strategy offers a number of advantages, chief among which is the ability to use legacy codes in each subdomain. However, it poses a number of challenges, especially in the presence of stochastic noise generated by the particle-based solver [3]. These include reduction in the order of accuracy of the constitutive solvers [22] and the need to propagate the noise throughout the whole simulation domain by adding a stochastic term to the (otherwise deterministic) continuum nonlinear solver [2,4,21,25]. Boundary conditions, in particular of Neumann and Robin types, pose another challenge to particle-based methods in general (e.g., [17] and the references therein), and to their use in hybrid simulations. (Even such a well-studied discrete model as Brownian motion appears to be rigorously analyzed and numerically implemented only on domains with reflecting boundaries [11] or, at steady state, witeDmann boundaries [15]; we are not aware of similar analyses of other discrete methods on bounded domains with general boundary conditions.)

In hybrid simulations, the difficulty of enforcing inhomogeneous boundary conditions can be obviated by replacing the coupling interface Γ with a "handshake region" wherein both discrete and continuous models are solved [1,7,21]. An added benefit of this approach is that it provides an iteration-free coupling. Disadvantages are the increased computational cost associated with having to solve the discrete model on a subdomain enlarged by at least one element of the continuum solver's discretization, and the high frequency of inter-solver communications. We therefore adopt a tight-coupling iterative strategy that both preserves the sharp interface Γ and reduces the computational cost of discrete (Monte Carlo) simulations. These two features are achieved by employing the reverse local Brownian motion (rBm) [12], a Monte Carlo approach that allows one to enforce inhomogeneous Dirichlet, Neumann, or Robin boundary conditions and is trivially parallelizable.

The theory and our implementation of rBm are discussed in Appendix B. In section 2 we formulate a hybrid discrete-continuum algorithm, which couples rBm and a diffusion equation, and discuss alternative coupling strategies in section 3. Numerical strategies for improving the accuracy of rBm in the presence of inhomogeneous Neumann boundary conditions are presented in section 3.5. The convergence, stability, and relative performance of our hybrid algorithm vis-à-vis the hybrid algorithm [1] based on standard (i.e., forward and global) Brownian motion are investigated in section 4. Major conclusions drawn from this analysis are collated in section 5.

2. Formulation of a hybrid model of diffusion

Consider a d-dimensional Lipschitz domain $D \subset \mathbb{R}^d$ consisting of two non-overlapping Lipschitz subdomains D_c and D_d on which continuum and discrete models of diffusion, respectively, are defined; the interface between D_c and D_d is denoted by Γ. Furthermore, let ∂D_c and ∂D_d denote the boundaries of D_c and D_d, respectively, such that $\partial D_\mathrm{c} \cap \partial D_\mathrm{d} = \Gamma$. A continuum model consists of a diffusion equation

$$\frac{\partial u_\mathrm{c}}{\partial t} = \alpha_\mathrm{c} \nabla^2 u_\mathrm{c}, \qquad \mathbf{x} \in D_\mathrm{c}, \quad t > 0 \tag{1a}$$

subject to initial and boundary conditions

$$u_\mathrm{c}(\mathbf{x}, 0) = u_\mathrm{in}, \quad \mathbf{x} \in D_\mathrm{c}; \tag{1b}$$

$$u_\mathrm{c}(\mathbf{x}, t) = u_\mathrm{D}, \quad \mathbf{x} \in \Gamma_\mathrm{D}; \qquad \mathbf{n} \cdot \nabla u(\mathbf{x}, t) = J_\mathrm{N}, \quad \mathbf{x} \in \Gamma_\mathrm{N} \tag{1c}$$

where α_c is the diffusion coefficient in the subdomain Ω_c; $u_\mathrm{in}(\mathbf{x})$ is an initial distribution of the state variable (e.g., concentration) $u(\mathbf{x}, t)$; and $u_\mathrm{D}(\mathbf{x}, t)$ and $J_\mathrm{N}(\mathbf{x}, t)$ are sufficiently smooth functions prescribed on the Dirichlet (Γ_D) and Neumann (Γ_N) boundary segments, respectively.

Brownian motion provides a discrete model of diffusion in the subdomain D_d. A particle's trajectory $\mathbf{X}(t)$ evolves in time according to a stochastic differential equation $\mathrm{d}\mathbf{X}(t) = \sqrt{2\alpha_\mathrm{d}}\,\mathrm{d}\mathbf{W}(t)$, where α_d is the diffusion coefficient in the subdomain Ω_d and $\mathrm{d}\mathbf{W}(t) \sim \mathcal{N}(0, \mathrm{d}t)$ is a d-dimensional Wiener process. Our Monte Carlo simulations use the rBm implementation, in which individual trajectories of N_MC particles released at point $\mathbf{x} \in D_\mathrm{d}$ at time t satisfy

$$\mathbf{X}(t - \Delta t_\mathrm{d}) = \mathbf{X}(t) - \sqrt{2\alpha_\mathrm{d}}\,\mathcal{N}(0, \Delta t_\mathrm{d}). \tag{2}$$

The fixed time step Δt_d is defined such that $t_n = t - n\Delta t_\mathrm{d}$ ($n = 0, 1, 2, \ldots$). The i-th particle ($i = 1, \ldots, N_\mathrm{MC}$) follows a random trajectory $\mathbf{X}_i(t_n)$ with $\mathbf{X}_i(t) = \mathbf{x}$ until either it reaches the Dirichlet boundary Γ_D at exit time $T_i = \inf\{0 < s < t : \mathbf{X}_i(s) \notin D_\mathrm{d}\}$ or $n\Delta t_\mathrm{d} = t$, whichever happens first. Hence, the set of N_MC trajectories consists of mutually exclusive subsets Λ_D and Λ_in, comprising N_D particles that satisfy the first stopping condition and N_in particles that satisfy the second stopping condition, respectively. If a particle reaches the Neumann boundary Γ_N before any of the two stopping conditions are met, it is reflected back along the normal to Γ_N inside the domain at a distance $\mathrm{d}h \sim \sqrt{2\alpha_\mathrm{d}\Delta t_\mathrm{d}}$. These N_N particles

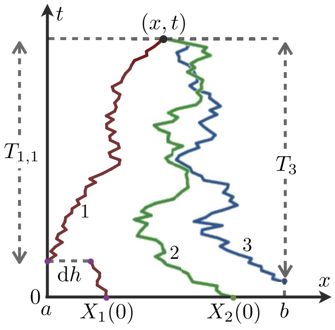

Fig. 1. Three particles, released at a space–time point (x, t), undergo one-dimensional reverse Brownian motion (rBm) along trajectories $X_i(t_n)$ with $i = 1, 2.3$ and $t_n = t - n\Delta t_\text{d}$ $(n = 0, 1, 2, \ldots)$. Deterministic Neumann and Dirichlet boundary conditions $\partial_x u = J_\text{N}(t)$ and $u = u_\text{D}(t)$ are prescribed at $x = a$ and $x = b$, respectively; a deterministic initial condition is $u(x, 0) = u_\text{in}(x)$. An rBm estimate of the solution of a diffusion equation, $\partial_t u = \alpha_\text{d}\partial_x^2 u$, at point (x, t) is $\hat{u}(x, t) = J_\text{N}(t - T_{1,1})\text{d}h + u_\text{in}[X_1(0)] + u_\text{in}[X_2(0)] + u_\text{D}(t - T_3)$.

form a set $\Lambda_\text{N} \in (\Lambda_\text{D} \cup \Lambda_\text{in})$. Given the initial and boundary functions $u_\text{in}(\mathbf{x})$, $u_\text{D}(\mathbf{x}, t)$ and $J_\text{N}(\mathbf{x}, t)$ specified for the discrete simulations domain, we compute the sample mean $\hat{u}_\text{d}(\mathbf{x}, t)$ of the random state variable $u_\text{d}(\mathbf{x}, t)$ at space–time point $(\mathbf{x}, t)$ as a weighted sum (see [12] and Appendix B)

$$\hat{u}_\text{d}(\mathbf{x}, t) = \frac{N_\text{in}}{N_\text{MC}}\hat{\mathcal{S}}_\text{in} + \frac{N_\text{D}}{N_\text{MC}}\hat{\mathcal{S}}_\text{D} + \frac{N_\text{N}}{N_\text{MC}}\hat{\mathcal{S}}_\text{N} \tag{3a}$$

of sample averages

$$\hat{\mathcal{S}}_\text{in} \equiv \frac{1}{N_\text{in}}\sum_{i\in\Lambda_\text{in}} u_\text{in}[\mathbf{X}_i(0)], \quad \hat{\mathcal{S}}_\text{D} \equiv \frac{1}{N_\text{D}}\sum_{i\in\Lambda_\text{D}} u_\text{D}[\mathbf{X}_i(t - T_i), t - T_i], \tag{3b}$$

$$\hat{\mathcal{S}}_\text{N} \equiv \frac{\sqrt{2\alpha_\text{d}\Delta t_\text{d}}}{N_\text{N}}\sum_{i\in\Lambda_\text{N}}\sum_{j=1}^{n_i} J_\text{N}[\mathbf{X}_i(t - T_{i,j}), t - T_{i,j}], \tag{3c}$$

where n_i is the number of times the ith particle reaches Γ_N, and $T_{i,j}$ is the jth reflection time of the ith particle. The simulation process is illustrated in Fig. 1. The sample means $\hat{\mathcal{S}}_\text{in}$, $\hat{\mathcal{S}}_\text{D}$, $\hat{\mathcal{S}}_\text{N}$ and, hence, $\hat{u}_\text{d}(\mathbf{x}, t)$ are different from the exact (ensemble mean) counterparts due to the finite number of samples. Confidence intervals for these estimates are derived in Appendix A.

These deterministic and stochastic problems are coupled at the interface Γ by enforcing the continuity conditions

$$u_\text{c}(\mathbf{x}, t) = \hat{u}_\text{d}(\mathbf{x}, t) \quad \text{and} \quad \alpha_\text{c}\mathbf{n}(\mathbf{x}) \cdot \nabla u_\text{c}(\mathbf{x}, t) = \alpha_\text{d}\mathbf{n}(\mathbf{x}) \cdot \nabla\hat{u}_\text{d}(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma, \tag{4}$$

where $\mathbf{n}(\mathbf{x})$ is the unit vector normal to Γ.

Alternative numerical strategies for this continuum-discrete hybrid model of diffusion are explored below. Before doing that, several salient features of rBm (3) are worthwhile discussing. First, the rBm treatment of the initial and Dirichlet conditions in (3) is exact, in the sense that the sample mean $\hat{u}_\text{d}$ converges to the ensemble mean $\mathbb{E}[u]$, i.e., $\hat{u}_\text{d}(\mathbf{x}, t) \to \mathbb{E}[u(\mathbf{x}, t)]$, in the limit of $N_\text{MC} \to \infty$ [12, Sec. 6.2.1]. The treatment of Neumann boundaries, the third term in (3), is heuristic; improving its accuracy and numerical implementation are goals of this study. Second, the rBm is a local particle method, i.e., it allows one to compute $\hat{u}$ at selected points $\mathbf{x} \in \Omega_\text{d}$, e.g., at points $\mathbf{x}$ in the neighborhood of Γ that are necessary to approximate the gradient $\nabla\hat{u}_\text{d}$ in (4). This renders the rBm-based hybrid more efficient than its counterpart based on standard Brownian motion, because the discrete domain Ω_d is much smaller than the continuum domain Ω_c (often it is a single element of a numerical mesh in Ω_c) and, hence, can be assigned a value of $\hat{u}_\text{d}$ at a single point $\mathbf{x} \in \Omega_\text{d}$ or an average over a few points.

3. Coupling algorithms

We analyze two coupling strategies, which are referred to here as the Dirichlet coupling and the Dirichlet–Neumann coupling. The former enforces continuity of the state variable, giving rise to a Dirichlet boundary condition at the interface Γ, while the latter alternates between assigning Dirichlet and Neumann boundary conditions.

3.1. Dirichlet coupling

Fig. 2 provides a workflow for this coupling strategy. At each macro-step of the hybrid algorithm, Δt_h, a continuum solution at the interface Γ, $u_\text{c}(\mathbf{x}, t + \Delta t_\text{h})$ for $\mathbf{x} \in \Gamma$, is treated as a Dirichlet boundary condition for the particle domain Ω_d. Using (3) with $\hat{\mathcal{S}}_\text{N} \equiv 0$, an rBm estimate of the mean solution, $\hat{u}_\text{d}(\mathbf{y}, t + \Delta t_\text{h})$, is computed only in a small ε-neighborhood of

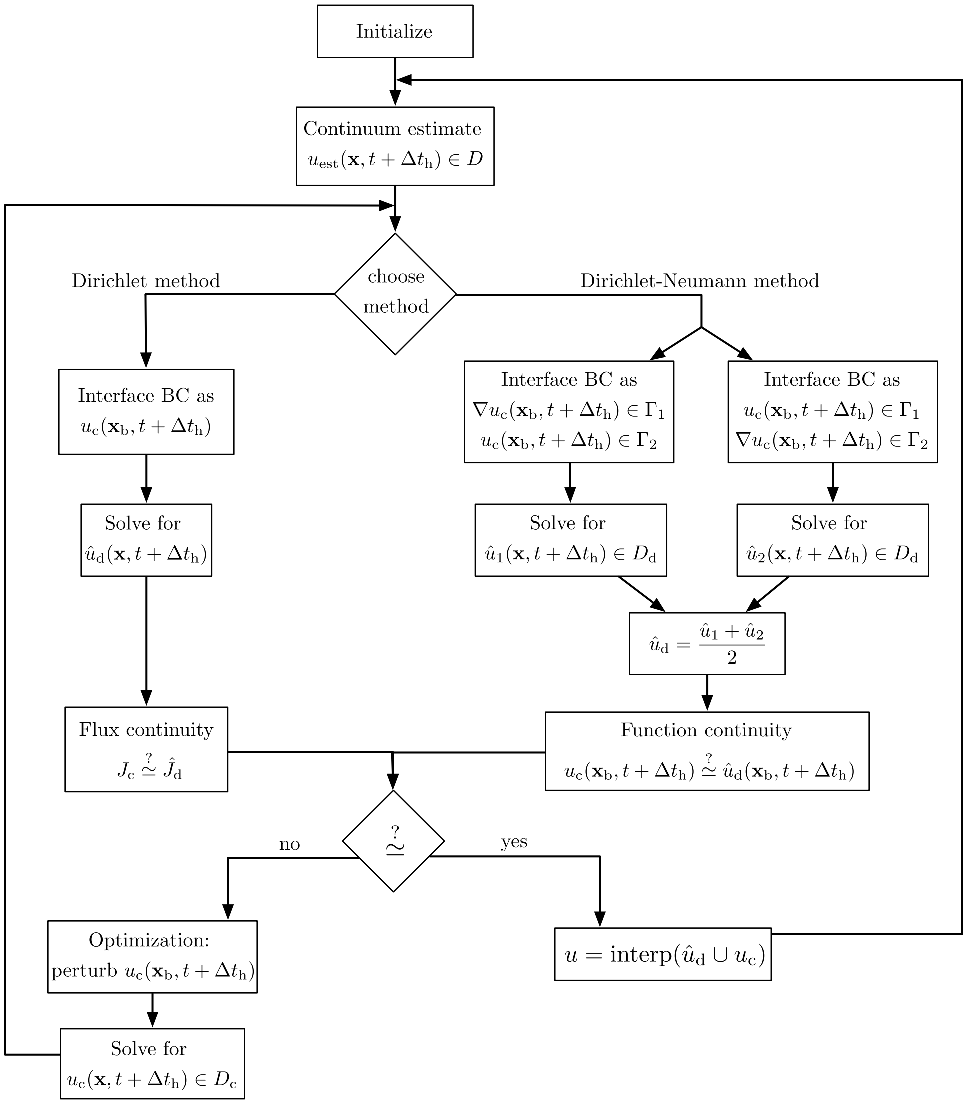
Initialize
Continuum estimate
$u_{\mathrm{est}}(\mathbf{x}, t+\Delta t_{\mathrm{h}}) \in D$
choose method
Dirichlet method
Dirichlet-Neumann method
Interface BC as
$u_{\mathrm{c}}(\mathbf{x}_{\mathrm{b}}, t+\Delta t_{\mathrm{h}})$
Interface BC as
$\nabla u_{\mathrm{c}}(\mathbf{x}_{\mathrm{b}}, t+\Delta t_{\mathrm{h}}) \in \Gamma_1$
$u_{\mathrm{c}}(\mathbf{x}_{\mathrm{b}}, t+\Delta t_{\mathrm{h}}) \in \Gamma_2$
Interface BC as
$u_{\mathrm{c}}(\mathbf{x}_{\mathrm{b}}, t+\Delta t_{\mathrm{h}}) \in \Gamma_1$
$\nabla u_{\mathrm{c}}(\mathbf{x}_{\mathrm{b}}, t+\Delta t_{\mathrm{h}}) \in \Gamma_2$
Solve for
$\hat{u}_{\mathrm{d}}(\mathbf{x}, t+\Delta t_{\mathrm{h}})$
Solve for
$\hat{u}_1(\mathbf{x}, t+\Delta t_{\mathrm{h}}) \in D_{\mathrm{d}}$
Solve for
$\hat{u}_2(\mathbf{x}, t+\Delta t_{\mathrm{h}}) \in D_{\mathrm{d}}$
$\hat{u}_{\mathrm{d}} = \frac{\hat{u}_1 + \hat{u}_2}{2}$
Flux continuity
$J_{\mathrm{c}} \overset{?}{\simeq} \hat{J}_{\mathrm{d}}$
Function continuity
$u_{\mathrm{c}}(\mathbf{x}_{\mathrm{b}}, t+\Delta t_{\mathrm{h}}) \overset{?}{\simeq} \hat{u}_{\mathrm{d}}(\mathbf{x}_{\mathrm{b}}, t+\Delta t_{\mathrm{h}})$
$\overset{?}{\simeq}$
no
yes
Optimization:
perturb $u_{\mathrm{c}}(\mathbf{x}_{\mathrm{b}}, t+\Delta t_{\mathrm{h}})$
$u = \mathrm{interp}(\hat{u}_{\mathrm{d}} \cup u_{\mathrm{c}})$
Solve for
$u_{\mathrm{c}}(\mathbf{x}, t+\Delta t_{\mathrm{h}}) \in D_{\mathrm{c}}$

Fig. 2. Workflow for the Dirichlet coupling strategy. Here u_{d} and u_{c} are the solutions in the discrete (D_{d}) and continuum (D_{c}) domains, and $\mathbf{x}_{\mathrm{b}}$ is a point of the interface Γ.

Γ, $V_\varepsilon = \{\mathbf{y} \in \Omega_{\mathrm{d}} : \|\mathbf{x} - \mathbf{y}\| \le \varepsilon \text{ for } \mathbf{x} \in \Gamma\}$. This solution is used to numerically approximate the interfacial flux $\hat{J}_{\mathrm{d}}(\mathbf{x}, t + \Delta t_{\mathrm{h}}) = -\alpha_{\mathrm{d}} \mathbf{n} \cdot \nabla \hat{u}_{\mathrm{d}}$. If the latter satisfies the second continuity condition in (4) within a prescribed tolerance, then the continuum model (1) provides an adequate representation of diffusion throughout the computational domain during time interval Δt_{h}, i.e., no hybrid simulations are required. Otherwise, the discrete and continuum interfacial fluxes thus computed are discontinuous, and enforcement of (4) necessitates an adjustment of the value of the hybrid solution u at the interface Γ, leading to an iterative procedure shown in Fig. 2.

Since this coupling strategy imposes only the Dirichlet boundary conditions at the interface Γ, it enables an exact treatment of the discrete problem with either standard or reverse Brownian motion. However, it calls for a derivative-free optimization method to achieve flux continuity at the interface, which might require a large number of iterations to converge.

3.2. Alternating Dirichlet–Neumann coupling

The coupling strategy shown in Fig. 2 ameliorates this problem at the cost of having to deal with a Neumann boundary condition. At each macro-step Δt_{h}, a numerical solution of the continuum model provides an initial guess of both $u(\mathbf{x}, t + \Delta t_{\mathrm{h}})$ and $J(\mathbf{x}, t + \Delta t_{\mathrm{h}}) = -\alpha_{\mathrm{d}} \mathbf{n} \cdot \nabla u$ at the interfacial points $\mathbf{x} \in \Gamma$. We use rBm to solve two problems: the first prescribes $u(\mathbf{x}, t + \Delta t_{\mathrm{h}})$ as a boundary condition on Γ, while the second imposes $J(\mathbf{x}, t + \Delta t_{\mathrm{h}})$ as a boundary condition on Γ. Let $\hat{u}_1(\mathbf{x}, t)$ and $\hat{u}_2(\mathbf{x}, t)$ denote the mean solutions of these two problems, respectively; both are computed with (3). (The computational cost of solving these two problems is essentially the same as that of solving one of them, since $\hat{u}_1$ and $\hat{u}_2$ are obtained by post-processing the same trajectories of the N_{MC} Brownian particles.) An interfacial average of these solutions, $u_{\text{av}}(\mathbf{x}, t) = (\hat{u}_1 + \hat{u}_2)/2$ for $\mathbf{x} \in \Gamma$, serves as a new Dirichlet boundary condition for the continuum problem. Its solution yields a new estimate of J, and the two discrete problems are solved again. This iterative procedure continues until $|\hat{u}_1 - \hat{u}_2| \le \epsilon_{\text{iter}}$ for all $\mathbf{x} \in \Gamma$, where ϵ_{iter} is a prescribed tolerance.

This Dirichlet–Neumann coupling strategy involves fewer optimization variables than the Dirichlet coupling method and, as we demonstrate below, proves to be more robust.

3.3. Algorithmic complexity

The `Continuum()` subroutine (line 5 in Algorithm 1) requires a runtime of $\mathcal{O}(N_{\text{grid}})$, where N_{grid} is the number of grid points in the continuum domain and the solution at each grid point is computed at a given time. The `Particle_sim()` subroutine (line 6) involves a runtime of $\mathcal{O}(N_{\text{MC}} \Delta t_{\text{h}} / \Delta t_{\text{d}})$, where $\Delta t_{\text{h}} / \Delta t_{\text{d}} \gg 1$. Finally, the `rBm()` subroutine (line 8) computes the estimate $\hat{u}_{\text{d}}$ by averaging over all particle positions $\mathbf{X}(T)$, $\mathcal{O}(N_{\text{MC}})$. Thus, the **while** loop requires $\mathcal{O}\left(K(N_{\text{grid}} + N_{\text{MC}})\right)$ computations, where K is the number of iterations required for convergence at each coupling time step Δt_{h}. Note that a naive implementation of the optimization procedure would perform both `Particles_sim()` and `rBm()` inside the **while** loop, such that particle trajectories are recomputed at every iteration, resulting in $\mathcal{O}\left(K\left(N_{\text{grid}} + N_{\text{MC}} \Delta t_{\text{h}} / \Delta t_{\text{d}}\right)\right)$ runtime.

Algorithm 1 Discrete-continuum hybrid algorithm.

```
 1: procedure HYBRID
 2:     Initialize: IC and BC
 3:     for each time index $i$ in $u[\mathbf{x} \in \Gamma, i\Delta t_{\text{h}}]$ do
 4:         IC $\equiv u(\mathbf{x}, (i-1)\Delta t_{\text{h}})$
 5:         Continuum estimate: $u_{\text{est}} =$ Continuum($u[\mathbf{x} \in D_{\text{c}}, i\Delta t_{\text{h}} | u_{\text{D}}$, IC])
 6:         Particle trajectories: $\mathbf{X}(T) =$ Particle_sim($\mathbf{X} \in D_{\text{d}}$)
 7:         while $|J_{\text{c}} - \hat{J}_{\text{d}}| > \epsilon_{\text{iter}}$ with optimization variable $u_\Gamma$ do
 8:             Compute $\hat{u}_{\text{d}}$ near $\Gamma$: $\hat{u}(\mathbf{x}, i\Delta t_{\text{h}}) =$ rBm($\mathbf{X}(T), u_\Gamma$, IC) for $\mathbf{x} \in V_\varepsilon$
 9:             Compute $u_{\text{c}}$: $u_{\text{c}}(\mathbf{x} \in D_{\text{c}}, i\Delta t_{\text{h}})$, given $u_\Gamma$ and IC
10:             Compute interfacial fluxes $J_{\text{c}}$ and $\hat{J}_{\text{d}}$
11:             Perturb($u_\Gamma$)
12:         end while
13:         $u(\mathbf{x}, n\Delta t_{\text{h}}) \leftarrow$ interpolate($u_{\text{c}} \cup \hat{u}_{\text{d}}$)
14:     end for
15: end procedure
```

Finally, the complexity of the `Perturb()` subroutine (line 11) depends on the iterative optimization procedure of u_Γ given by

$$u_\Gamma^* = \underset{u_\Gamma}{\operatorname{argmin}} \, |J_{\text{c}} - \hat{J}_{\text{d}}|.$$

In our numerical experiments, we perform the perturbation using the optimization MATLAB function `fminunc()`, which results in $K \sim \mathcal{O}(1)$.

Comparison of standard and reverse Brownian motion. Let N_{sBm} denote the number of particles used in standard (global and forward) Brownian motion (sBm) simulations. Then an sBm estimate $\tilde{u}(\mathbf{x}, t)$ is given by the fraction of N_{sBm} particles found in a volume $V(\mathbf{x})$ centered around $\mathbf{x}$ at time t, i.e., $\tilde{u}(\mathbf{x}, t)$ depends on the relative size of the averaging volume, $\|V\|/\|D_{\text{d}}\|$. This is in contrast to the rBm estimate $\hat{u}(\mathbf{x}, t)$, which has no averaging volume at all. This fact complicates a general comparison of the computational efficiency of the sBm and rBm simulations. However, the rBm approach has a definite edge in the context of hybrid modeling because it allows one to compute $\hat{u}(\mathbf{x}, t)$ directly at points of interest (e.g., in the ε-neighborhood of the interface Γ), while the sBm approach yields $\tilde{u}(\mathbf{x}, t)$ at every point (element) of the computational domain D_{d} and hence requires a larger number of MC simulations, $N_{\text{sBm}} \gg N_{\text{MC}}$.

Furthermore, a time step of the sBm algorithm is limited by the cell size $\|V\|$, while the rBm places no such constraint on the magnitude of Δt_{d}. The sBm method also requires smaller communication time-steps Δt_{h}, which are limited by the frequency with which particles are injected near the boundary in order to impose boundary conditions. Alternative strategies for employing large inter-solver communication time-steps Δt_{h} in the rBm method are discussed in section 3.4.

These computational advantages of rBm stem from its ability to explicitly account for the prescribed initial and boundary conditions. If sBm is implemented with a deterministic initial condition $u(\mathbf{x}, 0) = u_{\text{in}}(\mathbf{x})$, the "perfect" knowledge of $u_{\text{in}}(\mathbf{x})$ is lost when the particles are generated at the initialization stage. In rBm, the ensemble of N_{MC} particles explicitly accounts for $u_{\text{in}}(\mathbf{x})$. This is usually sufficient, since initial and boundary conditions are typically specified on average rather than by position of particles.

Finally, the computational cost of rBm depends on the mean exit time of Brownian particles. The latter it significantly shorter than that of sBm because $\hat{u}$ is computed only in the vicinity of the interface Γ, which reduces a particle's mean exit time $\mathbb{E}(T) = \bar{T}$ from the domain D_{d}. In one spatial dimension, i.e., for $D_{\text{d}} = [a, b]$, the latter is given by [10]

$$\bar{T}(x) = \frac{(x-a)(b-x)}{2\alpha_{\mathrm{d}}}, \qquad x \in (a,b). \tag{5}$$

Since the rBm-based hybrid simulations require an estimate $\hat{u}(x,t)$ close to the coupling interface, e.g., at $x = a + \varepsilon$, the particles' mean exit time is $\bar{T}(a+\varepsilon) \approx \varepsilon(b-a)/(2\alpha_{\mathrm{d}})$. This suggests that the most computationally intensive part of our hybrid algorithm, `Particle_sim()`, is reduced by a factor of $\bar{T}(a+\varepsilon)/t_{\mathrm{sim}}$. That results in a runtime of $\mathcal{O}\left((\bar{T}(a+\varepsilon)/t_{\mathrm{sim}})N_{\mathrm{MC}}\Delta t_{\mathrm{h}}/\Delta t_{\mathrm{d}}\right)$, where t_{sim} is the particle simulation horizon. In the context of hybrid simulations, $t_{\mathrm{sim}} = \Delta t_{\mathrm{h}}$, which yields a running time of $\mathcal{O}(\varepsilon C N_{\mathrm{MC}}/\Delta t_{\mathrm{d}})$, where $C = (b-a)/(2\alpha_{\mathrm{d}})$.

3.4. Strategies for reducing the inter-solver communication frequency

Any two-solver hybrid has three time scales: the time steps of its discrete (Δt_{d}) and continuum (Δt_c) components, and the frequency ($1/\Delta t_{\mathrm{h}}$) with which these solvers have to communicate with each other. Given the high computational cost of coupling algorithms, one would like to use a relatively large inter-solver communication time (macro-step), $\Delta t_{\mathrm{h}} \gg \max\{\Delta t_{\mathrm{c}}, \Delta t_{\mathrm{d}}\}$, whose size is independent from the components' time steps (typically, $\Delta t_{\mathrm{c}} \gg \Delta t_{\mathrm{d}}$).[1]

Let the inter-solver communication time be a multiple of the continuum solver's time step, $\Delta t_{\mathrm{h}} = n\Delta t_{\mathrm{c}}$ with $n \in \mathbb{N}$. Suppose that the continuity conditions (4) are enforced at both t and $t + \Delta t_{\mathrm{h}}$, such that $u_{\mathrm{c}}(\mathbf{x}_{\mathrm{b}}, t+\Delta t_{\mathrm{h}}) = \hat{u}_{\mathrm{d}}(\mathbf{x}_{\mathrm{b}}, t+\Delta t_{\mathrm{h}}) \equiv u_\Gamma(t+\Delta t_h)$ at any point $\mathbf{x}_{\mathrm{b}}$ of the interface Γ. The continuum solver requires knowledge of the boundary condition $u_\Gamma(\mathbf{x}_{\mathrm{b}}, t + k\Delta t_{\mathrm{c}})$ at intermediate times $k\Delta t_{\mathrm{c}} \le \Delta t_{\mathrm{h}}$ as well. We estimate its values via a linear interpolation,

$$u_\Gamma(t + k\Delta t_{\mathrm{c}}) = u_\Gamma(t) + \frac{u_\Gamma(t+\Delta t_{\mathrm{h}}) - u_\Gamma(t)}{\Delta t_{\mathrm{h}}} k\Delta t_{\mathrm{c}}, \tag{6}$$

for $k = 0, 1, \ldots, \Delta t_{\mathrm{h}}/\Delta t_{\mathrm{c}}$. For an ith particle used in the discrete simulations, (6) gives rise to

$$u_\Gamma(t + \Delta t_{\mathrm{h}} - T_i) = u_\Gamma(t+\Delta t_{\mathrm{h}}) - \frac{u_\Gamma(t+\Delta t_{\mathrm{h}}) - u_\Gamma(t)}{\Delta t_{\mathrm{h}}} T_i. \tag{7}$$

A similar interpolation is used for the continuum and discrete interfacial fluxes by replacing u_Γ with J_Γ in (6) and (7), respectively.

3.5. Improved accuracy of rBm on bounded domains

At each time step Δt_{d}, position $\mathbf{X}(t) = (X_1, \ldots, X_d)^\top$ of a particle undergoing the rBm changes to $\mathbf{X}(t - \Delta t_{\mathrm{d}})$ in accordance with (2). The probability of the particle crossing the domain's boundary ∂D_{d} exactly at exit time $T = \inf\{s > 0 : \mathbf{X}(s) \notin D_{\mathrm{d}}\}$ is $\mathbb{P}[\mathbf{X}(T) \in \partial D_{\mathrm{d}}] = 0$. Consequently, evaluation of the boundary functions u_{D} and J_Γ in (3) entails an error of order $\sqrt{2\alpha_{\mathrm{d}}\Delta t_{\mathrm{d}}}$.

When the discrete simulations domain Ω_{d} is fully embedded into the continuum simulations domain Ω_{d}, i.e., when $\partial D_{\mathrm{d}} = \Gamma$, this problem is resolved by using the continuum solution as a *boundary condition* to the discrete solution $\hat{u}_{\mathrm{d}}[\mathbf{X}_i(T_i)]$ at point $\mathbf{x} = \mathbf{X}_i$ and stopping time $T_i = \inf\{0 < s < t : \mathbf{X}_i(s) \in D_{\mathrm{c}}\}$ as soon as the ith Brownian particle enters the continuum domain D_{c}. This step requires storing the continuum solution $u_{\mathrm{c}}(\mathbf{x}, t)$ at each discrete time step Δt_{d} and at each iteration in order to interpolate the Neumann and Dirichlet boundary conditions in the space–time domain $D_{\mathrm{c}} \times [t, t+\Delta t_{\mathrm{h}}]$. This coupling strategy provides the most accurate solution in a discrete-to-continuum hybrid method (Fig. 3) but requires extensive memory storage and access time. Moreover, this method cannot deal with discrete domains that are not fully embedded in the continuum domain and with particles that might jump across the continuum domain, depending on the distance between Γ and Γ_{D}. Therefore, we address more general approximation strategies to increase the rBm solution's accuracy in the presence of the external Dirichlet (Γ_{D}) and Neumann (Γ_{N}) boundary conditions.

Since $\mathbb{P}[\mathbf{X}(T) \in \partial D_{\mathrm{d}}] = 0$, one has to either approximate $\hat{u}_{\mathrm{d}}[\mathbf{X}(T), T]$ or change the Brownian motion in a way that forces particles to end up exactly on the boundary ∂D_{d} at the exit time T. The former can be achieved by either performing a Taylor expansion around the boundary or redefining the conditions for which particles exit the domain. The latter can be done by restricting the motion of the Brownian particles to a grid. The method used in our simulations is described below, while alternative methods are discussed in Appendix C. Grid-based methods will be explored in the future.

Linear extrapolation for boundary overshoot. The use of discrete time steps in computer simulations implies that an ith particle crosses a Dirichlet boundary Γ_{D} at a discrete exit time $\check{T} = m\Delta t_{\mathrm{d}}$, where the integer m lies in the interval $1 \le m < \Delta t_{\mathrm{h}}/\Delta t_{\mathrm{d}}$. In other words, the ith particle overshoots, almost surely, the boundary ∂D_{d} by a distance $|\boldsymbol{\epsilon}_{\mathbf{x}}| = \mathcal{O}(\sqrt{2\alpha_{\mathrm{d}}\Delta t_{\mathrm{d}}})$ with the difference between the actual exit time T_i and its discrete counterpart $\check{T}$ denoted by $\epsilon_T = |T_i - \check{T}|$. Since T_i is between the last time the particle was inside D_{d} and the first time it ended up outside, $(m-1)\Delta t_{\mathrm{d}} < T_i < m\Delta t_{\mathrm{d}}$, ϵ_T is computed in rBm simulations from the knowledge of the particle positions right before and right after the exit,

[1] This is in contrast to the handshake-region-based coupling that employs either $\Delta t_{\mathrm{h}} = \Delta t_{\mathrm{c}}$ or $\Delta t_{\mathrm{h}} = \Delta t_{\mathrm{d}}$ (e.g., [1]).

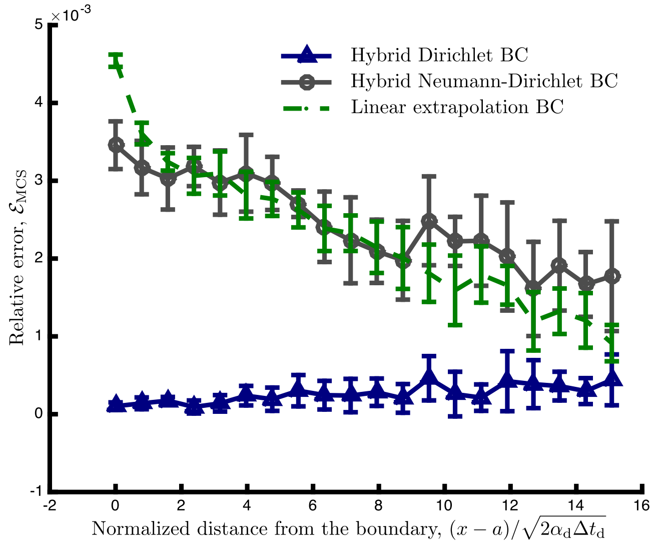

Fig. 3. Comparison of the boundary correction methods in terms of the relative error of solution estimates, $\mathcal{E}_{\mathrm{MCS}} = |1 - u_{\mathrm{MCS}}/u_{\mathrm{ex}}|$, at time $t = 4$ s (for $\alpha_{\mathrm{d}} = 0.17$ cm^2/s).

$$\epsilon_T = \frac{|\mathbf{X}_i(t - m\Delta t_{\mathrm{d}}) - \mathbf{x_b}|}{|\mathbf{X}_i(t - m\Delta t_{\mathrm{d}}) - \mathbf{X}_i(t - (m-1)\Delta t_{\mathrm{d}})|}\Delta t_{\mathrm{d}} \tag{8}$$

where $\mathbf{x_b}$ is the point of intersection between the boundary ∂D_{d} and the (line) trajectory of the particle upon exiting the domain.

If the ith particle exits Ω_{d} through the Dirichlet boundary segment Γ_{D}, then the value this particle acquires, i.e., $u_{\mathrm{d}}[\mathbf{X}_i(t - T_i), t - T_i] = u_{\mathrm{D}}(\mathbf{x}_{\mathrm{b}}, t - T_i)$ in (3b), is replaced with a first-order Taylor expansion of the boundary conditions in the vicinity of Γ_{D} around the time $t - T_i$,

$$\begin{aligned} u_{\mathrm{d}}(\mathbf{X}_i(t - \check{T}_i), t - \check{T}_i) &= u_{\mathrm{d}}(\mathbf{X}_i(t - T_i - \epsilon_T), t - T_i + \epsilon_T) \\ &\approx u_{\mathrm{D}}(\mathbf{x}_{\mathrm{b}}, t - T_i) + \epsilon_T \frac{Du_{\mathrm{d}}}{Dt}(\mathbf{x}_{\mathrm{b}}, t - T_i) \\ &= u_{\mathrm{D}}(\mathbf{x}_{\mathrm{b}}, t - T_i) + \epsilon_T \left[\frac{\partial u_{\mathrm{D}}}{\partial t} + \nabla u_{\mathrm{d}} \cdot \frac{\mathrm{d}\mathbf{X}_i}{\mathrm{d}t}\right]_{(\mathbf{x}_{\mathrm{b}},\, t - T_i)}. \end{aligned} \tag{9}$$

The gradient $\nabla u_{\mathrm{d}}(\mathbf{x_b}, t - T_i)$ and the boundary condition $u_{\mathrm{D}}(\mathbf{x_b}, t - T_i)$ are computed by linear interpolation, with (7). The particle velocity $\mathbf{v} = \mathrm{d}\mathbf{X}_i/\mathrm{d}t$ is calculated as $\mathbf{v} = [\mathbf{X}_i(t - m\Delta t_{\mathrm{d}}) - \mathbf{X}_i(t - (m-1)\Delta t_{\mathrm{d}})]/\Delta t_{\mathrm{d}}$. Since $\epsilon_T \mathbf{v} = \boldsymbol{\epsilon}_{\mathbf{x}}$, a correction for the Dirichlet boundary's contribution in (3) is

$$u_{\mathrm{d}}(\mathbf{X}_i(t - \check{T}_i), t - \check{T}_i) = u_{\mathrm{D}}(\mathbf{x_b}, t - T_i) + \boldsymbol{\epsilon}_{\mathbf{x}} \cdot \nabla u_{\mathrm{d}}(\mathbf{x_b}, t - T_i) + \epsilon_T \frac{\partial u_{\mathrm{D}}}{\partial t}. \tag{10}$$

A similar procedure applied to the Neumann boundaries yields a correction

$$J_{\mathrm{d}}(\mathbf{X}_i(t - \check{T}_i), t - \check{T}_i) = J_{\mathrm{N}}(\mathbf{x}_{\mathrm{b}}, t - T_i) + \boldsymbol{\epsilon}_{\mathbf{x}} \cdot \nabla J_{\mathrm{d}}(\mathbf{x}_{\mathrm{b}}, t - T_i) + \epsilon_{\mathrm{t}} \frac{\partial J_{\mathrm{N}}}{\partial t}. \tag{11}$$

Neumann boundary conditions. The empirical treatment of inhomogeneous Neumann boundary conditions in rBm (3c) calls for the particles that reach the boundary Γ_{N} to be reflected back inside the domain D_{d} by a distance $\delta_{\mathbf{x}}$. In the absence of its theoretical estimate, we conduct a series of numerical experiments to estimate an optimal value of $\delta_{\mathbf{x}}$.

The experiments involve rBm simulations of a one-dimensional diffusion equation $\partial_t u = \alpha_{\mathrm{d}} \partial_x^2 u$ on the interval $[0, 2\pi]$, subject to deterministic initial conditions $u_{\mathrm{in}}(x) = 2 + \sin(x)$ and Neumann and Dirichlet boundary conditions $J_{\mathrm{N}}(0, t) = 0.5$ and $u_{\mathrm{D}}(2\pi, t) = 2$, respectively. The large number of particles, $N_{\mathrm{MC}} = 50000$, renders the MC sampling error negligible.

Fig. 4 demonstrates the effect of the normalized reflection distance $\delta_x^\star = \delta_x/\sqrt{2\alpha_{\mathrm{d}}\Delta t_{\mathrm{d}}}$ on the relative error of the mean solution, $\mathcal{E}_{\mathrm{rBm}} = |1 - \hat{u}(x, t)/u_{\mathrm{ex}}(x, t)|$, where $u_{\mathrm{ex}}(x, t)$ is the exact solution. The error $\mathcal{E}_{\mathrm{rBm}}$ increases with $\delta_x^\star$ (Fig. 4a) and decreases with the distance x from the Neumann boundary Γ_{N} (Fig. 4b).

While using small values of $\delta_x^\star$ improves the rBm solution's accuracy, it significantly increases the computational cost (Fig. 5). That is because a reflected particle has a higher probability of hitting the boundary Γ_{N} again for a smaller reflection distance δ_x. Consequently, the size of Λ_{N} increases as δ_x decreases and more computations are required to find $\hat{\mathcal{S}}_{\mathrm{N}}$ in (3c). Therefore, we chose an optimal value of δ_x by minimizing a cost function

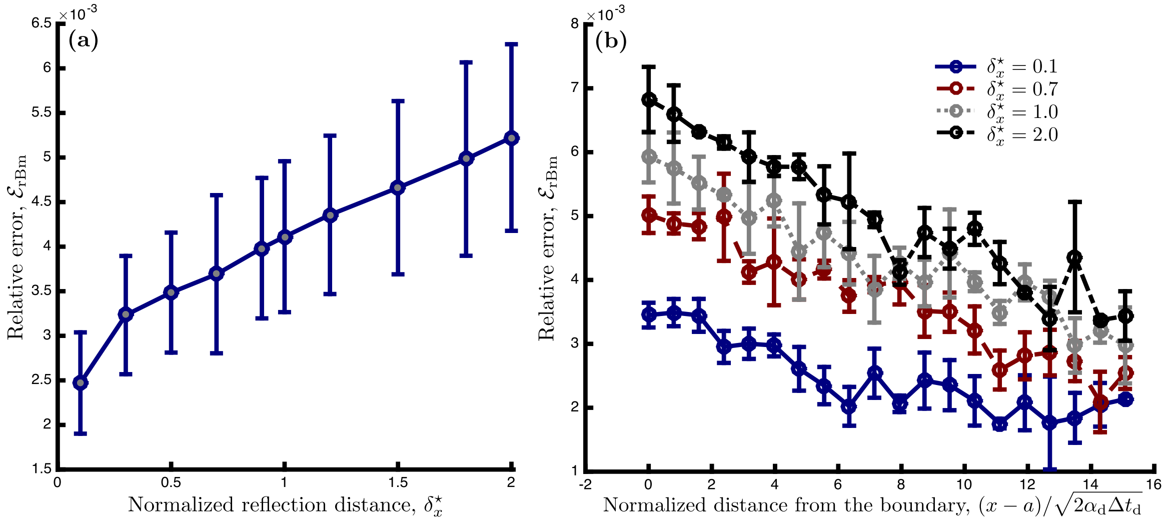

Fig. 4. Relative error in the rBm solution of the one-dimensional diffusion equation with an inhomogeneous Neumann boundary condition as a function of (a) normalized reflection distance $\delta_x^\star = \delta_x/\sqrt{2\alpha_{\mathrm{d}}\Delta t_{\mathrm{d}}}$ and (b) distance from the Neumann boundary.

$$C(\delta_x) = \mathcal{E}_{\mathrm{rBm}}(\delta_x) - \nu\mathcal{T}_{\mathrm{run}}(\delta_x) \tag{12}$$

where both the error $\mathcal{E}_{\mathrm{rBm}}(\delta_x)$ and the runtime $\mathcal{T}_{\mathrm{run}}(\delta_x)$ are obtained from Figs. 4a and 5a, respectively. The coefficient ν with dimension 1/s is chosen according to the available computational speed and the willingness to trade efficiency for accuracy.

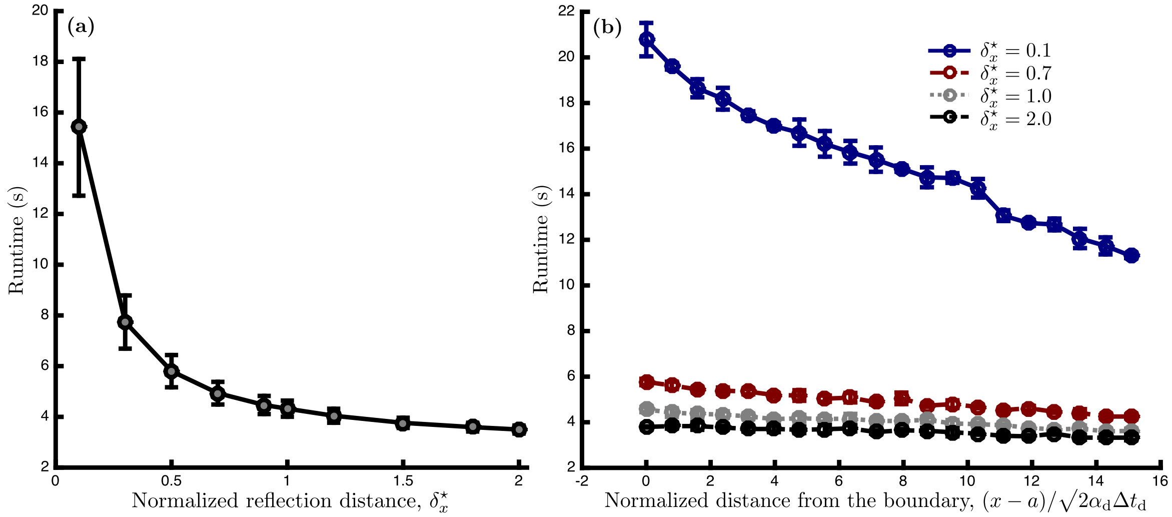

Fig. 5. Computational time for the rBm solution of one-dimensional diffusion equation with an inhomogeneous Neumann boundary condition as a function of (a) normalized reflection distance $\delta_x^\star = \delta_x/\sqrt{2\alpha_{\mathrm{d}}\Delta t_{\mathrm{d}}}$ and (b) distance from the Neumann boundary.

4. Simulation results

While the hybrid algorithm described above is applicable to any spatial dimension, we conduct a series of one-dimensional numerical experiments. This allows us to analyze the method's accuracy, convergence rate, and computational efficiency vis-à-vis the hybrid method [1] that employs the handshake-region coupling. In addition to providing a convenient computational testbed, such one-dimensional discrete/continuum hybrid models of diffusion are routinely used to represent phenomena as diverse as transport across a cell membrane [20], brain tumors [23], multiscale behavior of complex materials [6], complex fluid flow [24,16], and heat transfer in nano-structures [14].

4.1. Computational testbed

Consider a one-dimensional diffusive process on the interval $D = [0, 2\pi]$. Diffusion is driven by the difference between an initial condition $u(x, 0) = u_{\text{in}}(x) = 2 + \sin(x)$ and Dirichlet boundary conditions $u(x, t) = u_{\text{D}} = 2$ at $x = 0$ and $x = 2\pi$. The subsequent experiments are compared with the analytical solution $u_{\text{ex}}(x, t) = \exp(-\alpha_{\text{d}} t) \sin x + 2$. The continuum model, i.e., the one-dimensional version of (1), is valid for all $x \in [0, 2\pi]$ except for a (small) sub-interval $D_{\text{d}} = [a, b]$ where its validity breaks down and the discrete model (3) is used instead. In the simulations reported below, we set $a = \pi/2$, $b = 3\pi/4$, and $\alpha_{\text{d}} = \alpha_{\text{c}} = 0.2$. These and other physical quantities are defined in consistent units.

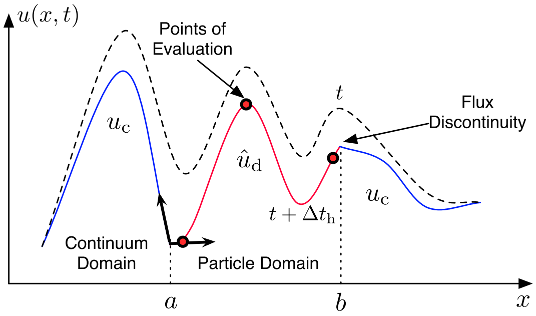

Fig. 6. The first step (pre-iteration) of the one-dimensional particle/continuum hybrid.

For small particle domains, the rBm-based Monte Carlo estimate $\hat{u}(x, t)$ has to be computed only at points $a_\delta = a + \delta$ and $b_\delta = b - \delta$ with $\delta \ll |b - a|$ (see Fig. 6), where it is required to estimate a sample mean of the interfacial fluxes, e.g., $\hat{J}_{\text{N}}(a, t) \approx -\alpha_{\text{d}}[\hat{u}(a + \delta, t) - \hat{u}(a, t)]/\delta$. In our simulations, however, the particle domain size is chosen to be large and we add a few more evaluation points in the particle domain in order to explore the interpolation error (line 13 in Algorithm 1). In the results reported below we set $\delta = \Delta x$, with the continuum domain's grid size $\Delta x = (b - a)/500$. The Monte Carlo simulations involve $N_{\text{MC}} = 5000$ particles.

4.2. Performance of the hybrid algorithm

A hybrid-simulations error $\mathcal{E}_{\text{hyb}} = \mathcal{E}_{\text{MCS}} + \mathcal{E}_{\text{PDE}} + \mathcal{E}_{\text{cpl}}$ consists of the rBm-based Monte Carlo simulations error $\mathcal{E}_{\text{MCS}}$, the discretization error $\mathcal{E}_{\text{PDE}}$ of the PDE solver, and the coupling error $\mathcal{E}_{\text{cpl}}$. The MCS error $\mathcal{E}_{\text{MCS}} = \mathcal{E}_{\text{sam}} + \mathcal{E}_{\text{rBm}}$, where $\mathcal{E}_{\text{sam}}$ is the sampling error due to the finite number of MC realizations N_{MC}, and $\mathcal{E}_{\text{rBm}}$ is the error introduced by the boundaries. The former is estimated by $\mathcal{E}_{\text{sam}} \sim 1/\sqrt{N_{\text{MC}}}$ according to the central limit theorem. The latter follows from the discussion in section 3.5 showing that $\mathcal{E}_{\text{rBm}}$, originally of $\mathcal{O}(\sqrt{\alpha_{\text{d}} \Delta t_{\text{d}}})$, is reduced to $\mathcal{O}(\alpha_{\text{d}} \Delta t_{\text{d}})$ by linear extrapolation (a first-order Taylor expansion truncation error).

The coupling error $\mathcal{E}_{\text{cpl}} = \mathcal{E}_{\text{com}}(\Delta t_{\text{h}}) + \mathcal{E}_{\text{int}}(\Delta t_{\text{h}}) + \mathcal{E}_{\text{tol}}$ arises from three independent sources. The communication error $\mathcal{E}_{\text{com}}$ is related to the number of times a particle and the continuum domain communicate via the flux continuity; this error increases with the hybrid time step Δt_{h}. The interpolation error $\mathcal{E}_{\text{int}}$ stems from the use of (6) and (7); as a truncation error of the first-order Taylor expansion it is $\mathcal{E}_{\text{int}} \sim \mathcal{O}(\Delta t_{\text{h}}^2)$. Finally, $\mathcal{E}_{\text{tol}} = |J_{\text{c}}(\mathbf{x}_{\text{b}} - \delta/2, t) - J_{\text{d}}(\mathbf{x}_{\text{b}} + \delta/2, t)|$ is the simulation tolerance for the algorithm's convergence.

Of these multiple sources of error only $\mathcal{E}_{\text{com}}(\Delta t_{\text{h}})$ and $\mathcal{E}_{\text{int}}(\Delta t_{\text{h}})$ are specific to hybrid simulations. We therefore study their impact on the hybrid error $\mathcal{E}_{\text{hyb}}$ as a function of Δt_{h}. Fig. 7 demonstrates that the error generated at the interface propagates from the discrete domain to the continuum domain and increases with Δt_{h}. However, the error at the interface is small independently of the size of Δt_{h}, which indicates that the optimal solution of u_Γ is attained at the end of each Δt_{h} and is unique. Figs. 7b and 8b show that $\mathcal{E}_{\text{cpl}} < 0.01$ even for the coupling time steps as large as 1 s. Finally, for the same error $\mathcal{E}_{\text{hyb}}$, the Dirichlet–Neumann method converges faster than the Dirichlet method.

Implementation recommendations. Given Δt_{d} and Δt_{c}, the simplest method of choosing Δt_{h} is $\Delta t_{\text{h}} = k\,\texttt{LCM}(\Delta t_{\text{d}}, \Delta t_{\text{c}})$, where $k \in \mathbb{N}$ (typically, $k = 1$) and LCM stands for Least Common Multiplier. However, in general, Δt_{h} should be chosen depending on the desired accuracy ($\mathcal{E}_{\text{hyb}}$) of the hybrid method. Therefore, we choose a range for Δt_{h} given a desired range of $\mathcal{E}_{\text{hyb}}$, and then tweak Δt_{d} and Δt_{c} such that $\texttt{LCM}(\Delta t_{\text{d}}, \Delta t_{\text{c}})$ falls in that range.

The accuracy of the Dirichlet–Neumann method and that of the Dirichlet method are similar. The differences are in the ease of implementation, robustness and the computational cost. The Dirichlet–Neumann method requires two solutions at every iteration while the Dirichlet method requires only one. However, our experiments suggest that the number of solves required until convergence is roughly the same for both methods; the Dirichlet method being a little faster when Δt_{h} is small and the initial guess close to the minimum (sometimes converging after only one iteration). This being said, the Dirichlet method requires the use of a nonlinear derivative-free optimization algorithm whose convergence properties depend on numerical implementation of the optimization procedure. In contrast, the Dirichlet–Neumann method updates

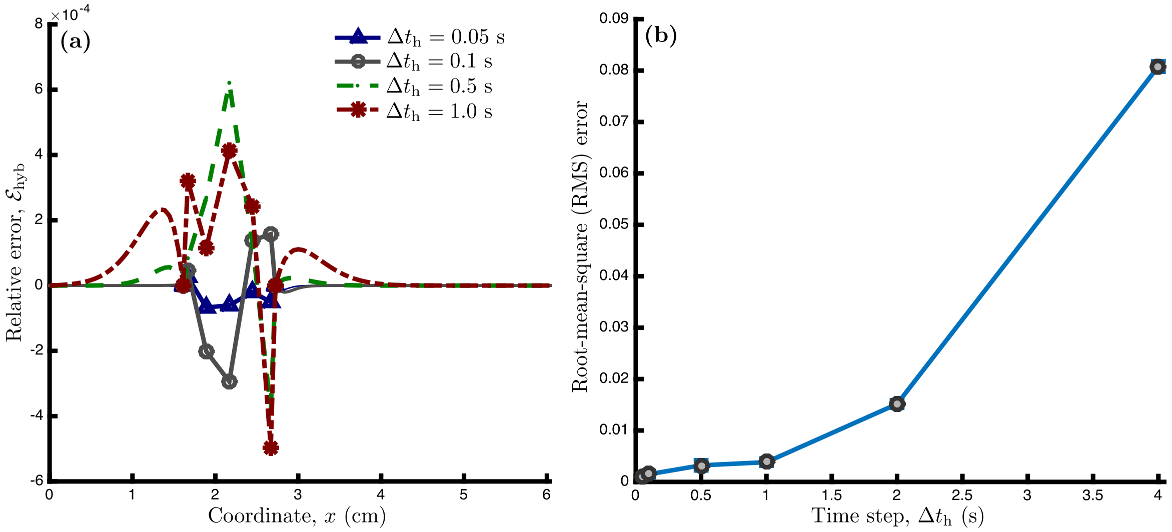

Fig. 7. Dirichlet–Neumann method: (a) relative error $\mathcal{E}_{\rm hyb}(x, t = 4 \text{ s}) = |u_{\rm hyb} - u_{\rm ex}|$ for several values of the hybrid time step $\Delta t_{\rm h}$ and (b) the corresponding RMS error as a function of $\Delta t_{\rm h}$.

u_Γ by averaging the two solutions that are computed at every iteration. Therefore, the Dirichlet–Neumann hybrid method is more robust in complex domains and has an optimization procedure that is easier to implement than the one used in the Dirichlet method. Given this tradeoff between robustness and efficiency, we recommend using the Dirichlet–Neumann method in problems that involve large coupling time steps $\Delta t_{\rm h}$ and rough solutions at the interface, and using the Dirichlet method for problems with a smoothly varying solution at the interface where stability is not a concern and computational speed is desired.

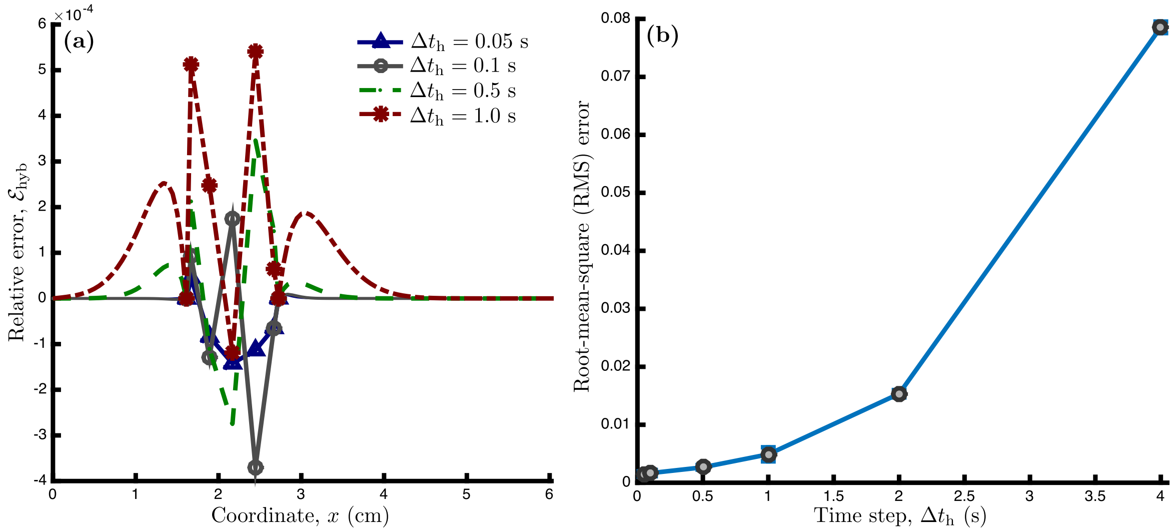

Fig. 8. Dirichlet method: (a) relative error $\mathcal{E}_{\rm hyb}(x, t = 4 \text{ s}) = |u_{\rm hyb} - u_{\rm ex}|$ for several values of the hybrid time step $\Delta t_{\rm h}$ and (b) the corresponding RMS error as a function of $\Delta t_{\rm h}$.

5. Conclusions

We developed an efficient and robust hybrid method to couple Brownian motion with a continuum diffusion equation. In the context of discrete-to-continuum hybrid simulations, the use of Monte Carlo simulations based on reverse Brownian motion (rBm) has a number of advantages. These include

1. the ability to use hybrid coupling time steps $\Delta t_{\rm h}$ that are as large as those used in continuum-to-continuum hybrid algorithms, i.e., larger than what is possible with standard Brownian motion [1];
2. the possibility of computing the solution only near the boundaries, where it is required to ensure the continuity of the flux in the hybrid method;

3. the ability to use the continuum domain as a deterministic source of the Dirichlet, Neumann and initial conditions for the rBm; and
4. a controllable loss of accuracy given a flexible choice of N_{MC} at every location in the particle domain.

Our hybrid algorithm is easy to implement in any number of dimensions. Furthermore, extending the hybrid model to advection–diffusion equations is relatively straightforward.

The performance of the proposed hybrid algorithm can be improved by

1. developing efficient methods to deal with fully embedded particle domains, e.g., by storing the continuum solution only near the interface, and
2. using a grid-based Brownian motion to reduce boundary errors.

Given the flexibility of the rBm method, our algorithm can be extended to hybrid methods with overlapping discrete and continuum domains, offering the ability to choose particle stopping times based on characteristics of the continuum solution (e.g., local uncertainty, measure of dissipation, etc.)

Acknowledgements

This work was supported in part by the Defense Advanced Research Projects Agency under the EQUiPS program, the Air Force Office of Scientific Research under Grant FA9550-12-1-0185 and by the National Science Foundation under Grant DMS-1522799.

Appendix A. Confidence intervals for Brownian motion

A confidence interval for the estimate $\hat{u}_{\mathrm{d}}(\mathbf{x},t)$ depends on $\hat{\mathcal{S}}_{\mathrm{in}}$, $\hat{\mathcal{S}}_{\mathrm{D}}$ and $\hat{\mathcal{S}}_{\mathrm{N}}$, which in turn depend on time t and the initial and boundary conditions. The estimates $\hat{\mathcal{S}}_{\mathrm{in}}$, $\hat{\mathcal{S}}_{\mathrm{D}}$ and $\hat{\mathcal{S}}_{\mathrm{N}}$ in (3) are sampled from $u_{\mathrm{in}}(\mathbf{x})$, $u_{\mathrm{D}}(\mathbf{x},t)$ and $J_{\mathrm{N}}(\mathbf{x},t)$ respectively, which are deterministic and known in advance. Therefore, the variances of $\mathcal{S}_{\mathrm{in}}$, $\mathcal{S}_{\mathrm{D}}$ and $\mathcal{S}_{\mathrm{N}}$ are given by

$$\sigma^2_{\mathcal{S}_{\mathrm{in}}}(\mathbf{x},t) = \int\limits_{D_{\mathrm{d}}} \frac{1}{\sqrt{4\pi Dt}} \mathrm{e}^{-\frac{(\mathbf{y}-\mathbf{x})^2}{4Dt}} \mathbf{y}^2 u_{\mathrm{in}}(\mathbf{y}) \mathrm{d}\mathbf{y}, \qquad \mathbf{x} \in D_{\mathrm{d}}; \tag{A.1}$$

$$\sigma^2_{\mathcal{S}_{\mathrm{D}}}(\mathbf{x},t) = \int\limits_0^t \int\limits_{\Gamma_{\mathrm{D}}} \tau^2 \mathbf{y}^2 u_{\mathrm{D}}(\mathbf{y},\tau) \mathrm{d}\mathbf{y}\mathrm{d}\tau, \qquad \mathbf{x} \in \Gamma_{\mathrm{D}}; \tag{A.2}$$

$$\sigma^2_{\mathcal{S}_{\mathrm{N}}}(\mathbf{x},t) = \int\limits_0^t \int\limits_{\Gamma_{\mathrm{N}}} \tau^2 \mathbf{y}^2 J_{\mathrm{N}}(\mathbf{y},\tau) \mathrm{d}\mathbf{y}\mathrm{d}\tau, \qquad \mathbf{x} \in \Gamma_{\mathrm{N}}. \tag{A.3}$$

The standard error for a given term $\mathcal{S}_i(\mathbf{x},t)$ is a random variable defined by

$$Z = \frac{\hat{\mathcal{S}}_i - \mathcal{S}_i}{\sigma_{\mathcal{S}_i}/\sqrt{N_i}}, \qquad i = \mathrm{in}, \mathrm{D}, \mathrm{N}, \tag{A.4}$$

where N_i is the sample size. We seek bounds on the exact mean $\mathcal{S}_i$, given a certain confidence level $(1-\alpha)$, such that

$$\mathbb{P}(-z^* < Z < z^*) = \mathbb{P}\Big(-z^* < \frac{\hat{\mathcal{S}}_i - \mathcal{S}_i}{\sigma_{\mathcal{S}_i}/\sqrt{N_i}} < z^*\Big) = 1-\alpha, \tag{A.5}$$

where z^* can be determined if the distribution function is known, e.g., Gaussian. Even though Z is generally non-Gaussian, the confidence interval ($\mathtt{CI}$) can be calculated, for large N_i, from the central limit theorem

$$\mathtt{CI}_i = \left[\hat{\mathcal{S}}_i - \frac{\sigma_{\mathcal{S}_i}}{\sqrt{N_i}} z^*, \hat{\mathcal{S}}_i + \frac{\sigma_{\mathcal{S}_i}}{\sqrt{N_i}} z^*\right]. \tag{A.6}$$

For a Monte Carlo simulation with N_{in}, N_{D} and N_{N}, the $\mathtt{CI}$ of $\hat{u}_{\mathrm{d}}(\mathbf{x},t)$ is given by

$$\mathtt{CI}_{\hat{u}_{\mathrm{d}}}(\mathbf{x},t) = \sum_i^{\mathrm{in,D,N}} \frac{N_i}{N_{\mathrm{MC}} + N_{\mathrm{N}}} \left[\hat{\mathcal{S}}_i - \frac{\sigma_{\mathcal{S}_i}}{\sqrt{N_i}} z^*, \hat{\mathcal{S}}_i + \frac{\sigma_{\mathcal{S}_i}}{\sqrt{N_i}} z^*\right]. \tag{A.7}$$

Unfortunately, N_{N} is a priori unknown. For Dirichlet boundary conditions, $N_{\mathrm{N}} = 0$ and the exit probability $\mathbb{P}_{\mathrm{e}}(\mathbf{x},t)$ can be obtained from a Fokker–Planck equation [10, Section 5.2.8]. In this case, the confidence interval is expressed as a function of N_{MC}:

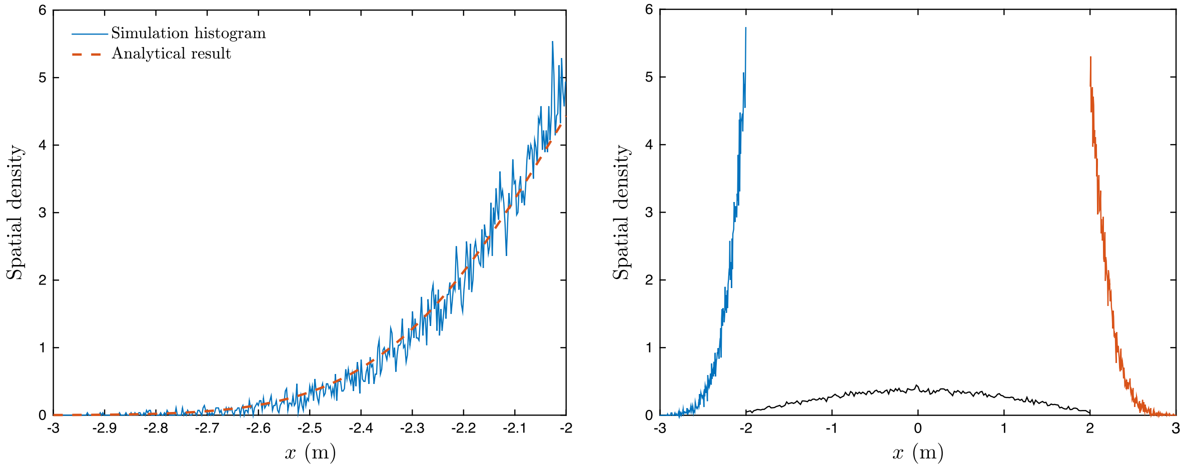

Fig. 9. Particle density distribution: (a) analytical vs experimental comparison on the left side of the domain, (b) density of particles throughout the whole domain, with probability to be exactly at the boundary $p(x_\mathrm{b},t)=0$.

$$\mathtt{CI}_{\hat{u}_\mathrm{d}}(\mathbf{x},t)=\mathbb{P}_\mathrm{e}\left[\hat{\mathcal{S}}_\mathrm{D}-\frac{\sigma_{\mathcal{S}_\mathrm{D}}z^*}{\sqrt{N_\mathrm{MC}\mathbb{P}_\mathrm{e}}},\ \hat{\mathcal{S}}_\mathrm{D}+\frac{\sigma_{\mathcal{S}_\mathrm{D}}z^*}{\sqrt{N_\mathrm{MC}\mathbb{P}_\mathrm{e}}}\right]+(1-\mathbb{P}_\mathrm{e})\left[\hat{\mathcal{S}}_\mathrm{in}-\frac{\sigma_{\mathcal{S}_\mathrm{in}}z^*}{\sqrt{N_\mathrm{MC}(1-\mathbb{P}_\mathrm{e})}},\ \hat{\mathcal{S}}_\mathrm{in}+\frac{\sigma_{\mathcal{S}_\mathrm{in}}z^*}{\sqrt{N_\mathrm{MC}(1-\mathbb{P}_\mathrm{e})}}\right]. \tag{A.8}$$

Appendix B. Particle-based simulations of diffusion

Consider a d-dimensional diffusion equation $\partial_t u=\alpha_\mathrm{d}\nabla^2 u$ defined on a bounded domain $D\in\mathbb{R}^d$, and subject to the initial and boundary conditions (1b). It can also be thought of as a Fokker–Planck or Smoluchowski equation for the probability density function of the position of particles undergoing Brownian motion. A particle trajectory $\mathbf{X}(t)$ satisfies a Langevin equation $\mathrm{d}\mathbf{X}=\sqrt{2\alpha_\mathrm{d}}\,\mathrm{d}\mathbf{W}(t)$, where $\mathrm{d}\mathbf{W}(t)\sim\mathcal{N}(0,\mathrm{d}t)$ is an $\mathbb{R}^d$-valued Wiener process.

According to [12], the reverse Brownian motion (rBm) accounts for time by adding an extra dimension with unit negative drift,

$$\tilde{\mathbf{X}}=(\mathbf{X},X_{d+1})^\top\in\mathbb{R}^{d+1}:\quad \mathrm{d}\mathbf{X}=\sqrt{2\alpha_\mathrm{d}}\,\mathrm{d}\mathbf{W}(s),\quad \mathrm{d}X_{d+1}(s)=-\mathrm{d}s, \tag{B.1}$$

giving rise to an $\mathbb{R}^{d+1}$ advection–diffusion process. In this formulation, the stopping time

$$T=\inf\{s>0:\mathbf{X}(s)\notin D_t\},\qquad D_t=D\times(0,t) \tag{B.2}$$

with initial conditions $X_{d+1}=t>0$ and $\mathbf{X}(0)=\mathbf{x}\in D$, represents the time of exit through either time or space "boundaries" (Fig. 1). Given the initial condition $u_\mathrm{in}(\mathbf{x})$ and Dirichlet boundary condition $u(\mathbf{x},t)=u_\mathrm{D}(\mathbf{x},t)$ on the domain's surface $\mathbf{x}\in\Gamma$, the mean solution $\hat{u}(\mathbf{x},t)$ of (B.1) is given by $\hat{u}(\mathbf{x},t)=\mathbb{E}^{(\mathbf{x},t)}[u(\mathbf{X}(T))]$. Recalling the definition of the stoppage time T in (B.2) this yields

$$\hat{u}(\mathbf{x},t)=\mathbb{E}^{(\mathbf{x},t)}[u_\mathrm{in}(\mathbf{X}(T))|T=t]\mathbb{P}(T=t)+\mathbb{E}^{(\mathbf{x},t)}[u_\mathrm{D}(\mathbf{X}(t-T))|T<t]\mathbb{P}(T<t) \tag{B.3}$$

where $\mathbb{E}$ and $\mathbb{P}$ denote the expectation and probability of a random event, respectively.

Appendix C. Alternate boundary correction methods

In this section, we propose probabilistic methods to deal with the rBm error due to the finite exit distance at the stopping time.

A statistical approach to account for the error is to use the exit distance distribution of particles, $f(\mathbf{x})$, shown in Fig. 9b, at time T and position $\mathbf{X}_i(T_i)$, such that $\mathbf{x}\in\overline{D}_\mathrm{d}\equiv\mathbb{R}^n\setminus D_\mathrm{d}$. In one dimension, $\overline{D}_\mathrm{d}=(-\infty,a)\cup(b,\infty)$. We define $f_a(x)$ for $x\in(-\infty,a)$ to be the left side density of particles upon exiting D_d. The distribution shown in Fig. 9a is computed as

$$f_a(x)=\int_a^b p(x,t;y,t-\Delta t_\mathrm{d})\mathrm{d}y \tag{C.1}$$

where $p(x,t;y,t-\Delta t_\mathrm{d})$ is the joint probability density that the Brownian particle is at y at time $t-\Delta t_\mathrm{d}$ and at x at time t. It is given by

$$p(x,t;y,t-\Delta t_\mathrm{d})=\frac{1}{\sqrt{4\pi\alpha_\mathrm{d}\Delta t_\mathrm{d}}}\mathrm{e}^{-\frac{(x-y)^2}{4\alpha_\mathrm{d}\Delta t_\mathrm{d}}}. \tag{C.2}$$

Therefore,

$$f_a(x) = \frac{1}{2}\left[\operatorname{erf}\left(\frac{x-a}{2\sqrt{\alpha_\mathrm{d}\Delta t_\mathrm{d}}}\right) - \operatorname{erf}\left(\frac{x-b}{2\sqrt{\alpha_\mathrm{d}\Delta t_\mathrm{d}}}\right)\right], \tag{C.3}$$

which is renormalized with $\int_{-\infty}^{a} f_a(x)\mathrm{d}x$ to obtain the corresponding probability density function of exit distance. Knowing the mean exit distance $\gamma = \int_{-\infty}^{a} x f_a(x)\mathrm{d}x$, we shift the original boundary a to a new position $a+\gamma$ inside the domain, where $\gamma = a - \mathbb{E}[X(T)|X(T) \leq a]$ so that the mean exit distance becomes a. At b, the exit distribution is $f_b(x) = (b-a) + f_a(-x)$ by symmetry. In higher dimensions, the mean exit distance γ can still be computed but depends on the geometry. Similarly, knowing the mean exit distance "envelope", we can shift the entire boundary inward to account for the error.

Without further analysis, we can assume that for large enough number of particles $N \to \infty$, the expected exit distance converges to $\mathbb{E}[\mathbf{X}(T)]$. This method can be extended to higher dimensions using the probability flux of the corresponding Fokker–Planck equation. This method requires a large number of particles in order to converge, which might be a limitation when a small number of particles is desired.

A similar solution can be obtained by simply perturbing the boundary according to the distribution $f(\mathbf{x})$. In this case, we perturb the boundary in such a way to get $\mathbb{E}[\mathbf{X}(T)|\mathbf{X} \notin D_\mathrm{d}] = \mathbf{x} \in \Gamma$. The perturbation of the boundary is done easily in 1D but requires variational methods for higher dimensions.

References

[1] F.J. Alexander, A.L. Garcia, D.M. Tartakovsky, Algorithm refinement for stochastic partial differential equations: I. Linear diffusion, J. Comput. Phys. 182 (1) (2002) 47–66.

[2] F.J. Alexander, A.L. Garcia, D.M. Tartakovsky, Algorithm refinement for stochastic partial differential equations: II. Correlated systems, J. Comput. Phys. 207 (2) (2005) 769–787.

[3] F.J. Alexander, A.L. Garcia, D.M. Tartakovsky, Noise in algorithm refinement methods, Comput. Sci. Eng. 7 (3) (2005) 32–38.

[4] J.B. Bell, J. Foo, A.L. Garcia, Algorithm refinement for the stochastic Burgers' equation, J. Comput. Phys. 223 (1) (2007) 451–468.

[5] T.-J. Choi, M.R. Maurya, D.M. Tartakovsky, S. Subramaniam, Stochastic operator-splitting method for reaction–diffusion systems, J. Chem. Phys. 137 (18) (2012) 184102.

[6] W.A. Curtin, R.E. Miller, Atomistic/continuum coupling in computational materials science, Model. Simul. Mater. Sci. Eng. 11 (3) (2003) R33.

[7] A. Donev, J.B. Bell, A.L. Garcia, B.J. Alder, A hybrid particle-continuum method for hydrodynamics of complex fluids, Multiscale Model. Simul. 8 (3) (2010) 871–911.

[8] T.O. Drews, E.G. Webb, D.L. Ma, J. Alameda, R.D. Braatz, R.C. Alkire, Coupled mesoscale-continuum simulations of copper electrodeposition in a trench, AIChE J. 50 (1) (2004) 226–240.

[9] A.L. Garcia, J.B. Bell, W.Y. Crutchfield, B.J. Alder, Adaptive mesh and algorithm refinement using direct simulation Monte Carlo, J. Comput. Phys. 154 (1) (1999) 134–155.

[10] C.W. Gardiner, Stochastic Methods: A Handbook for the Natural and Social Sciences, 4th edition, Springer, Berlin, 2010.

[11] D.S. Grebenkov, Residence times and other functionals of reflected Brownian motion, Phys. Rev. E 76 (2007) 041139.

[12] M. Grigoriu, Stochastic Calculus: Applications in Science and Engineering, Springer Science & Business Media, 2002.

[13] D.E. Keyes, L.C. McInnes, C. Woodward, et al., Multiphysics simulations: challenges and opportunities, Int. J. High Perform. Comput. Appl. 27 (1) (2013) 4–83.

[14] J. Liu, S. Chen, X. Nie, M.O. Robbins, A continuum–atomistic simulation of heat transfer in micro- and nano-flows, J. Comput. Phys. 227 (1) (2007) 279–291.

[15] S. Maire, E. Tanré, Monte Carlo approximations of the Neumann problem, Monte Carlo Methods Appl. 19 (3) (2013) 201–236.

[16] S.T. O'Connell, P.A. Thompson, Molecular dynamics/continuum hybrid computations: a tool for studying complex fluid flows, Phys. Rev. E 52 (6) (1995) R5792–R5795.

[17] E.M. Ryan, A.M. Tartakovsky, C. Amon, A novel method for modeling Neumann and Robin boundary conditions in smoothed particle hydrodynamics, Comput. Phys. Commun. 181 (12) (2010) 2008–2023.

[18] M. Salloum, K. Sargsyan, R. Jones, B. Debusschere, H.N. Najm, H. Adalsteinsson, A stochastic multiscale coupling scheme to account for sampling noise in atomistic-to-continuum simulations, Multiscale Model. Simul. 10 (2) (2012) 550–584.

[19] T. Schwartzentruber, L. Scalabrin, I. Boyd, A modular particle-continuum numerical method for hypersonic non-equilibrium gas flows, J. Comput. Phys. 225 (1) (2007) 1159–1174.

[20] W. Stein, Transport and Diffusion Across Cell Membranes, Elsevier, 2012.

[21] S. Taverniers, F.J. Alexander, D.M. Tartakovsky, Noise propagation in hybrid models of nonlinear systems: the Ginzburg–Landau equation, J. Comput. Phys. 262 (2014) 313–324.

[22] S. Taverniers, A.Y. Pigarov, D.M. Tartakovsky, Conservative tightly-coupled simulations of stochastic multiscale systems, J. Comput. Phys. 313 (2016) 400–414.

[23] Z. Wang, T.S. Deisboeck, Computational modeling of brain tumors: discrete, continuum or hybrid?, Sci. Model. Simul. 15 (1–3) (2008) 381–393.

[24] H.S. Wijesinghe, R.D. Hornung, A.L. Garcia, N.G. Hadjiconstantinou, Three-dimensional hybrid continuum–atomistic simulations for multiscale hydrodynamics, J. Fluids Eng. 126 (5) (2004) 768–777.

[25] S.A. Williams, J.B. Bell, A.L. Garcia, Algorithm refinement for fluctuating hydrodynamics, Multiscale Model. Simul. 6 (4) (2008) 1256–1280.