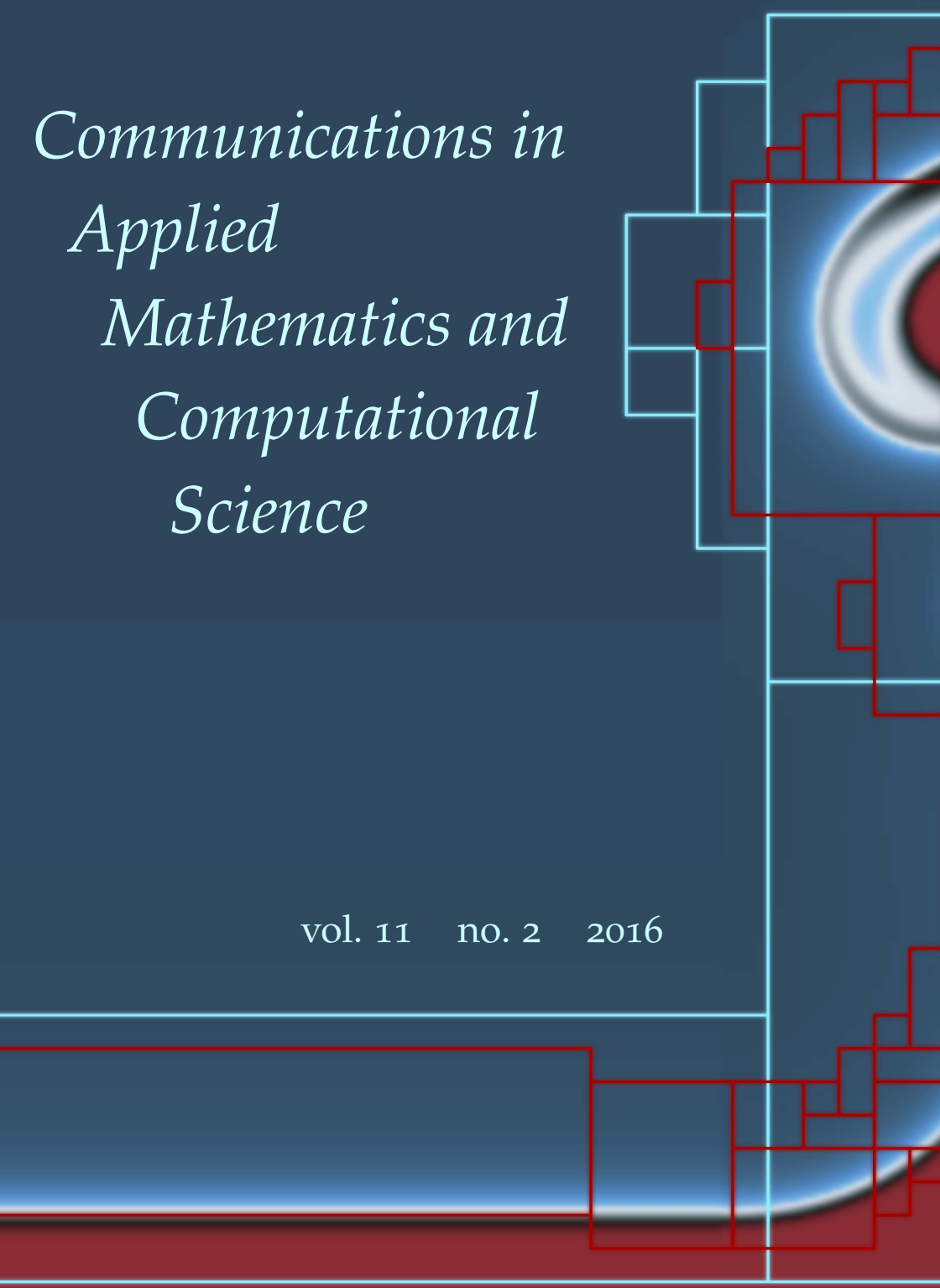
Communications in Applied Mathematics and Computational Science

vol. 11 no. 2 2016



msp

Communications in Applied Mathematics and Computational Science

msp.org/camcos

See inside back cover or msp.org/camcos for submission instructions.

The subscription price for 2016 is US $95/year for the electronic version, and $135/year (+$15, if shipping outside the US) for print and electronic. Subscriptions, requests for back issues from the last three years and changes of subscribers address should be sent to MSP.

Communications in Applied Mathematics and Computational Science (ISSN 2157-5452 electronic, 1559-3940 printed) at Mathematical Sciences Publishers, 798 Evans Hall #3840, c/o University of California, Berkeley, CA 94720-3840, is published continuously online. Periodical rate postage paid at Berkeley, CA 94704, and additional mailing offices.

CAMCoS peer review and production are managed by EditFLOW® from MSP.

PUBLISHED BY

mathematical sciences publishers

nonprofit scientific publishing

http://msp.org/

COMM. APP. MATH. AND COMP. SCI.
Vol. 11, No. 2, 2016
dx.doi.org/10.2140/camcos.2016.11.143

A REAL-SPACE GREEN'S FUNCTION METHOD FOR THE NUMERICAL SOLUTION OF MAXWELL'S EQUATIONS

BORIS LO, VICTOR MINDEN AND PHILLIP COLELLA

A new method for solving the transverse part of the free-space Maxwell equations in three dimensions is presented. By taking the Helmholtz decomposition of the electric field and current sources and considering only the divergence-free parts, we obtain an explicit real-space representation for the transverse propagator that explicitly respects finite speed of propagation. Because the propagator involves convolution against a singular distribution, we regularize via convolution with smoothing kernels (B-splines) prior to sampling based on a method due to Beyer and LeVeque (1992). We show that the ultimate discrete convolutional propagator can be constructed to attain an arbitrarily high order of accuracy by using higher-order regularizing kernels and finite difference stencils and that it satisfies von Neumann's stability condition. Furthermore, the propagator is compactly supported and can be applied using Hockney's method (1970) and parallelized using the same observation as made by Vay, Haber, and Godfrey (2013), leading to a method that is computationally efficient.

1. Introduction

In this paper, we present a method for solving Maxwell's equations. Our approach will be based on the expression of the evolution of the magnetic and transverse electric fields in terms of a first-order, constant-coefficient hyperbolic system

$$\begin{aligned} \frac{\partial \boldsymbol{u}(\boldsymbol{x},t)}{\partial t} &= L\boldsymbol{u}(\boldsymbol{x},t) + \boldsymbol{f}(\boldsymbol{x},t), \quad (\boldsymbol{x},t) \in \mathbb{R}^D \times \mathbb{R}_+, \\ \boldsymbol{u}(\boldsymbol{x},0) &= \boldsymbol{u}_0(\boldsymbol{x}), \qquad\qquad\qquad \boldsymbol{x} \in \mathbb{R}^D, \end{aligned} \tag{1}$$

where L is a constant-coefficient first-order linear differential operator in space and $\boldsymbol{f}$ is some known source term. Formally, the solution to (1) can be written explicitly using Duhamel's formula:

$$\boldsymbol{u}(\boldsymbol{x},t+\Delta t) = e^{L\Delta t}\boldsymbol{u}(\boldsymbol{x},t) + \int_0^{\Delta t} e^{L(\Delta t - s)}\boldsymbol{f}(\boldsymbol{x},t+s)\,ds. \tag{2}$$

MSC2010: primary 65M12, 65M80; secondary 65D05, 65D07, 78M25.
Keywords: Maxwell's equations, Green's function, high order.

This is the starting point for a broad class of time-integration schemes, known as exponential integrators [10], that use (2) to eliminate stiff terms contained in L, which, if treated explicitly, would impose unnecessary and undesirable time-step constraints. This constraint is removed by applying (2) to the spatially discretized equations and evaluating the action of the matrix corresponding to $e^{\eta L}$ on a vector using fast matrix-free methods. Such methods eliminate the stability constraint corresponding to the fast time scales in L.

In the present work, we use (2) as a starting point for eliminating the speed-of-light CFL stability condition in solving Maxwell's equations by explicitly discretizing an integral form of the propagator $e^{\eta L}$ for the original system of PDEs. This type of approach has been proposed previously in [2] and further examined in [13; 9]. In our approach, we use a Helmholtz decomposition to treat the divergence-free and curl-free parts of the solution separately, which allows us to express the propagator in terms of convolutions with weighted delta distributions over the sphere $|\boldsymbol{x}| = c\eta$, where c is the speed of light. We then discretize (2) in space by replacing the delta distributions with regularized approximate delta distributions defined on a rectangular grid, using the ideas in [19]. This leads to approximations of any order of spatial accuracy of the continuous propagator by discrete convolution operators on a rectangular grid. The discrete kernel satisfies a form of finite propagation speed; i.e., its support is contained in a bounded set of grid points of radius $\mathcal{O}(\sigma + P)$, where σ is the CFL number for the speed of light and P is the order of accuracy of the spatial approximation. This naturally leads to a domain-decomposition formulation of the problem, in which the convolution over the entire domain is replaced with a collection of convolutions over small patches that cover the domain. Due to boundedness of the support of the discrete propagator kernel, the resulting parallel application of the propagator is independent of the decomposition into patches. Finally, the evaluation of the time integral in (2) is approximated by quadratures, and the discrete convolutions are evaluated using FFTs with Hockney's method [11, pp. 180–181]. This method is closely related to the domain decomposition in [20] but differs from that method in that the starting point for our method is a discretization of a real-space propagator while the approach in [20] discretizes a propagator in Fourier space. We will discuss the relative merits of the two approaches in Section 6.

The remainder of this paper is organized as follows. In Section 2, we formalize our problem statement and present a high-level outline of our algorithm and its various components. In Section 3, we describe the discretization process in detail for a comprehensive presentation of each component of the algorithm. In Section 4, we perform a stability analysis of our procedure showing that under certain assumptions the von Neumann stability condition is satisfied. In Section 5, we present a number of numerical tests that show an implementation of our algorithm in action as

applied to the free-space Maxwell equations. Finally, in Section 6, we make some concluding remarks.

2. Problem statement and derivation of propagators

Here, and in what follows, functions of space and/or time will frequently be written omitting their explicit spatial and temporal dependencies, e.g., $\psi = \psi(\boldsymbol{x}, t)$. We will also use the operator notation $\mathcal{P}^t(\boldsymbol{u}) \equiv e^{Lt}(\boldsymbol{u})$. In this notation, (2) is written as

$$\boldsymbol{u}(t+\Delta t) = \mathcal{P}^{\Delta t}(\boldsymbol{u}_0) + \int_0^{\Delta t} \mathcal{P}^{\Delta t - s}(\boldsymbol{f}_s)\, ds, \tag{3}$$

where $f_s(t) \equiv f(t+s)$.

2.1. ***The scalar wave equation propagator.*** To illustrate our approach, we will first derive a real-space propagator for the 3-D wave equation,

$$\begin{gathered} \frac{\partial^2 \phi}{\partial t^2} = \Delta \phi, \\ \phi(\boldsymbol{x}, 0) = \phi_0(\boldsymbol{x}), \quad \frac{\partial \phi(\boldsymbol{x}, 0)}{\partial t} = \psi_0(\boldsymbol{x}). \end{gathered} \tag{4}$$

We introduce unknowns $\boldsymbol{v} \equiv \nabla\phi$ and $p \equiv \frac{\partial \phi}{\partial t}$, permitting us to recast (4) as a first-order hyperbolic system for $\boldsymbol{v}$ and p with initial-value constraints, i.e.,

$$\begin{gathered} \frac{\partial \boldsymbol{v}}{\partial t} = \nabla p, \qquad \frac{\partial p}{\partial t} = \nabla \cdot \boldsymbol{v}, \\ \boldsymbol{v}(\boldsymbol{x}, 0) = \boldsymbol{v}_0(\boldsymbol{x}) \equiv \nabla\phi_0(\boldsymbol{x}), \quad p(\boldsymbol{x}, 0) = p_0(\boldsymbol{x}) \equiv \psi_0(\boldsymbol{x}). \end{gathered} \tag{5}$$

Because $\boldsymbol{v}_0 = \nabla\phi$ is the gradient of some function, we see that $\nabla \times \boldsymbol{v}_0 \equiv 0$. This implies that $\nabla \times \boldsymbol{v}_t = 0$ for all time $t > 0$. The curl-free constraint on $\boldsymbol{v}_0$ is a necessary and sufficient condition for the first-order system (5) to be equivalent to (4) as it is necessary for v_0 to be curl-free so that the second-order equation can be recovered from the first-order system.

Taking the Fourier transform in $\boldsymbol{x}$, we obtain

$$\frac{\partial}{\partial t}\begin{bmatrix} \tilde{\boldsymbol{v}}(\boldsymbol{k}, t) \\ \tilde{p}(\boldsymbol{k}, t) \end{bmatrix} = \begin{bmatrix} 0 & i\boldsymbol{k} \\ i\boldsymbol{k}^T & 0 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{v}}(\boldsymbol{k}, t) \\ \tilde{p}(\boldsymbol{k}, t) \end{bmatrix}, \tag{6}$$

where we interpret the Fourier variable $\boldsymbol{k}$ as a column vector. The operator exponential of this system matrix scaled by Δt is $\widetilde{\mathcal{P}}_W^{\Delta t}$, the Fourier transform of our desired propagator. Since we need consider only curl-free $\boldsymbol{v}$, we see that $\widetilde{\mathcal{P}}_W^{\Delta t}$ can be written

$$\widetilde{\mathcal{P}}_W^{\Delta t} = \begin{bmatrix} \cos|\boldsymbol{k}|\Delta t & i\boldsymbol{k}(\sin|\boldsymbol{k}|\Delta t)/|\boldsymbol{k}| \\ i\boldsymbol{k}^T(\sin|\boldsymbol{k}|\Delta t)/|\boldsymbol{k}| & \cos|\boldsymbol{k}|\Delta t \end{bmatrix}. \tag{7}$$

Note that, in full generality, the top-left block of $\widetilde{\mathcal{P}}_W^{\Delta t}$ has terms involving $\hat{\boldsymbol{k}}\hat{\boldsymbol{k}}^T$ and $I - \hat{\boldsymbol{k}}\hat{\boldsymbol{k}}^T$, but the block reduces to $\cos|\boldsymbol{k}|\Delta t$ when restricted to curl-free input.

Taking an inverse Fourier transform and defining the kernels $G^{\Delta t}$ and $H^{\Delta t}$ via

$$G^{\Delta t}(z) \equiv \frac{\delta(|z| - \Delta t)}{4\pi\,\Delta t}, \tag{8}$$

$$H^{\Delta t}(z) \equiv \frac{\partial}{\partial s}\left(\frac{\delta(|z| - s)}{4\pi s}\right)\bigg|_{s=\Delta t}, \tag{9}$$

we see that the action of the propagator on an arbitrary state vector $\boldsymbol{h}(\boldsymbol{x}) \equiv \begin{bmatrix}\boldsymbol{f}(\boldsymbol{x}) & g(\boldsymbol{x})\end{bmatrix}^T$ with $\boldsymbol{f}$ curl-free is given by

$$\mathcal{P}_W^{\Delta t}(\boldsymbol{h}) = \begin{bmatrix} H^{\Delta t} * \boldsymbol{f} + G^{\Delta t} * \nabla g \\ G^{\Delta t} * (\nabla \cdot \boldsymbol{f}) + H^{\Delta t} * g \end{bmatrix}, \tag{10}$$

where convolutions are defined spatially as

$$(K * f)(\boldsymbol{x}) \equiv \int_{\mathbb{R}^3} K(\boldsymbol{y})\, f(\boldsymbol{x} - \boldsymbol{y})\, d\boldsymbol{y} \tag{11}$$

and convolution of a scalar quantity with a vector quantity is taken componentwise. Considering again (3) and noting the lack of sources, we see that we can obtain the solution to (5) for any final time $t_{\text{final}} = \Delta t$ by evaluating

$$\begin{bmatrix} \boldsymbol{v}(\Delta t) \\ p(\Delta t) \end{bmatrix} = \begin{bmatrix} H^{\Delta t} * \boldsymbol{v}_0 + G^{\Delta t} * \nabla p_0 \\ G^{\Delta t} * (\nabla \cdot \boldsymbol{v}_0) + H^{\Delta t} * p_0 \end{bmatrix}. \tag{12}$$

Here, the absence of sources obviates the need to treat the time integral in (3) and therefore reduces the problem entirely to discretely applying (10). We note that (10) can be derived from the classical solution starting with the second-order formulation as seen in [22]. However, we outline the approach starting with the first-order system as an analog to Maxwell's equations.

We see that for application of (10) it is necessary to approximate convolution against the singular kernels $G^{\Delta t}$ and $H^{\Delta t}$. To make $H^{\Delta t}$ more amenable to approximation, we use some calculus to reduce convolutions against $H^{\Delta t}$ to convolutions against $G^{\Delta t}$ combined with weights and spatial derivatives. We begin with the Fourier relationship

$$\frac{\partial}{\partial s}\left(\frac{\delta(|\boldsymbol{x}| - s)}{4\pi s}\right) = \mathcal{F}^{-1}[\cos|\boldsymbol{k}|s]. \tag{13}$$

It is not difficult to verify that if we write $\boldsymbol{x}$ and $\boldsymbol{k}$ in terms of their components then we have

$$\cos|\boldsymbol{k}|s = \frac{\sin|\boldsymbol{k}|s}{|\boldsymbol{k}|s} - i\sum_{d=1}^{3} \frac{\partial}{\partial k_d}\left(\frac{\sin|\boldsymbol{k}|s}{|\boldsymbol{k}|s}\right) i k_d. \tag{14}$$

Defining the new convolutional kernels $G_d^{\Delta t}$ for $d = 1, 2, 3$ via

$$G_d^{\Delta t}(z) \equiv \frac{z_d \delta(|z| - \Delta t)}{4\pi \Delta t}, \tag{15}$$

i.e., convolution in space against the weighted distribution $x_d \delta(|\boldsymbol{x}| - \Delta t)/(4\pi)$, we see from standard rules of Fourier analysis that (14) is the Fourier transform of the operator that acts on a function $f : \mathbb{R}^3 \mapsto \mathbb{R}$ via

$$(H^{\Delta t} * f) = \frac{1}{\Delta t} G^{\Delta t} * f - \sum_{d=1}^{3} G_d^{\Delta t} * \frac{\partial f}{\partial z_d}. \tag{16}$$

2.2. ***The Maxwell propagator.*** Similarly to Section 2.1, we can derive an expression for the propagator for the solution of Maxwell's equations written in terms of spatial derivatives and convolutions. We begin by writing the set of Maxwell's equations for $(\boldsymbol{x}, t) \in \mathbb{R}^3 \times \mathbb{R}_+$ as

$$\frac{\partial \boldsymbol{E}}{\partial t} = c\nabla \times \boldsymbol{B} - \boldsymbol{J}, \quad \frac{\partial \boldsymbol{B}}{\partial t} = -c\nabla \times \boldsymbol{E}, \tag{17}$$

$$\nabla \cdot \boldsymbol{E} = \rho, \qquad \nabla \cdot \boldsymbol{B} = 0, \tag{18}$$

with appropriate initial conditions. Here, $\boldsymbol{E}$ and $\boldsymbol{B}$ are the electric and magnetic fields, respectively, $\boldsymbol{J}$ is a known current source term, ρ is the bound current density, and c is the speed of light in vacuum.

To find a solution for Maxwell's equations, we first use a Helmholtz decomposition to break the electric field and current source into their longitudinal (curl-free) and transverse (divergence-free) parts

$$\boldsymbol{E} = \boldsymbol{E}_L + \boldsymbol{E}_T, \quad \boldsymbol{J} = \boldsymbol{J}_L + \boldsymbol{J}_T, \tag{19}$$

where $\nabla \times \boldsymbol{E}_L \equiv 0$ and $\nabla \cdot \boldsymbol{E}_T \equiv 0$ and similarly for $\boldsymbol{J}_L$ and $\boldsymbol{J}_T$. This decomposition leads to a first-order system of hyperbolic PDEs describing the coupling between $\boldsymbol{E}_T$ and $\boldsymbol{B}$ (see (17))

$$\frac{\partial \boldsymbol{E}_T}{\partial t} = c\nabla \times \boldsymbol{B} - \boldsymbol{J}_T, \quad \frac{\partial \boldsymbol{B}}{\partial t} = -c\nabla \times \boldsymbol{E}_T, \tag{20}$$

$$\nabla \cdot \boldsymbol{E}_T = 0, \qquad \nabla \cdot \boldsymbol{B} = 0. \tag{21}$$

The divergence-free conditions (21) are initial-value constraints, similar to the curl-free constraint on $\boldsymbol{v}_0$ for the wave equation, and if satisfied at time $t = 0$, then they are satisfied at all later times according to (20). The longitudinal component of the electric field $\boldsymbol{E}_L$ can be specified either in terms of Coulomb's law,

$$\boldsymbol{E}_L = -\nabla \phi, \quad \nabla \cdot \boldsymbol{E}_L = \rho, \tag{22}$$

or directly from applying the Helmholtz decomposition to the evolution equation for $\boldsymbol{E}$,

$$\frac{\partial \boldsymbol{E}_L}{\partial t} = -\boldsymbol{J}_L. \tag{23}$$

These two specifications of $\boldsymbol{E}_L$ are equivalent, provided that, at time $t = 0$, (22) is satisfied. In practice, the choice of which of these two formulations to use in discretizing $\boldsymbol{E}_L$ depends on the details of how the evolution of ρ is specified. The risk is that, by using (23), accumulation of numerical error will cause (22) not to be satisfied. We will not address this issue here other than to note that the formulation given here will require the solution of Poisson's equation at least to compute the Helmholtz decomposition of $\boldsymbol{J}$ and possibly to solve (22). For those problems, we can use fast Poisson solvers, the cost of which will be made up for by the ability to take larger time steps. Therefore, for the purposes of demonstrating the properties of the method described here, we will consider only examples in which $\boldsymbol{E}_L \equiv 0$ and $\boldsymbol{J}_L \equiv 0$. Given this and using reasoning similar to that in Section 2.1, we obtain the action of the Maxwell propagator on a state vector $\boldsymbol{h}(\boldsymbol{x}) = \left[\boldsymbol{E}_T(\boldsymbol{x}) \;\; \boldsymbol{B}(\boldsymbol{x})\right]^T$,

$$\mathcal{P}_M^{\Delta t}(\boldsymbol{h}) = \begin{bmatrix} H^{c\Delta t} * \boldsymbol{E}_T + G^{c\Delta t} * (\nabla \times \boldsymbol{B}) \\ -G^{c\Delta t} * (\nabla \times \boldsymbol{E}_T) + H^{c\Delta t} * \boldsymbol{B} \end{bmatrix}, \tag{24}$$

from which the action on a divergence-free current source can be inferred. Contrary to the source-free case we saw before, the appearance of $\boldsymbol{J}_T$ in (20) will require the treatment of the integral in (3) to obtain the full solution.

3. Discretization

As seen in Section 2, the wave equation and Maxwell propagators involve convolution against kernels taking the form of (possibly weighted) delta distributions supported on spheres. For example, convolution against the kernel $G_d^{\Delta t}$ in (15) is given by

$$(G_d^{\Delta t} * w)(\boldsymbol{x}) \equiv \int \frac{z_d}{4\pi\,\Delta t}\delta(|\boldsymbol{z}| - \Delta t)\, w(\boldsymbol{x} - \boldsymbol{z})\, d\boldsymbol{z} = \int_{\partial B_{\Delta t}} \frac{z_d}{4\pi\,\Delta t} w(\boldsymbol{x} - \boldsymbol{z})\, d\boldsymbol{z}, \tag{25}$$

where $\partial B_{\Delta t}$ is the 3-D sphere of radius Δt. Rewriting these convolutions as integrals over singular surfaces as we have done above, we see that accurately computing these convolutions reduces to accurately evaluating integrals of the form

$$I \equiv \int_\Gamma g(z)\, f(z)\, dz, \tag{26}$$

where Γ is a continuous and bounded surface and $g \in C(\mathbb{R}^d)$ is a weighting function.

Accurate discretization on a Cartesian grid of the integral I is treated succinctly by Tornberg and Engquist [19], who summarize a framework for replacing such

integrals with sums of samples of a regularized integrand based on work concerning singular source terms by Beyer and LeVeque [4]. We describe this in more detail below.

3.1. ***Regularized delta distributions.*** As a precursor to integration over multi-dimensional surfaces, consider integrating a function $f : \mathbb{R} \to \mathbb{R}$ against a 1-D delta distribution with arbitrary center $\bar{x} \in \mathbb{R}$, i.e., evaluating $f(\bar{x})$ via the sifting formula

$$f(\bar{x}) = \int_{\mathbb{R}} f(x)\delta(x - \bar{x})\, dx. \tag{27}$$

The above integral can be thought of as integrating $f(x)$ over a singular surface of dimension 0, supported at the single point $\bar{x}$. Because of its singular nature, simply sampling $f(x)\delta(x - \bar{x})$ on a grid and approximating integration with summation is not a numerically well defined operation. Instead, given a grid spacing h, we consider sampling a *regularized* approximant $\delta_h(x - \bar{x})$. Beyer and LeVeque [4] introduce the set of discrete moment conditions for such an approximant, summarized succinctly in [19].

Definition 1 (discrete moment conditions [4; 19]). Given a grid spacing $h > 0$, we say a function $\delta_h : \mathbb{R} \to \mathbb{R}$ is in the function class Q^q if δ_h has compact support $[-mh, mh]$ for some $m > 0$ and

$$h \sum_{j \in \mathbb{Z}} (jh - \bar{x})^r \delta_h(jh - \bar{x}) = \begin{cases} 1, & r = 0, \\ 0, & 1 \le r < q, \end{cases} \tag{28}$$

for any $\bar{x} \in \mathbb{R}$.

We note that the conditions in Definition 1 are simply discrete analogues of the continuous moment conditions

$$\int_{-\infty}^{\infty} (x - \bar{x})^r \delta(x - \bar{x})\, dx = \begin{cases} 1, & r = 0, \\ 0, & 1 \le r < q, \end{cases} \tag{29}$$

which are satisfied by the delta distribution for arbitrarily large q.

For sufficiently regular functions f, the discrete moment conditions are sufficient for consistency of δ_h; i.e., for $\delta_h \in Q^q$ and $f \in C^q$, we have the asymptotic error bound

$$f(\bar{x}) - h \sum_{j \in \mathbb{Z}} f(jh)\delta_h(jh - \bar{x}) = \mathcal{O}(h^q) \tag{30}$$

for any $\bar{x} \in \mathbb{R}$ [4]. In the Appendix, we give a review of how to find such 1-D approximants satisfying the discrete moment conditions to order q while attaining the minimum necessary support, which is essentially accomplished by piecewise Lagrange interpolation.

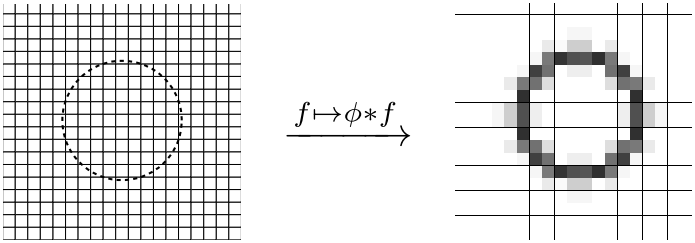


Figure 1. Left: a 2-D slice of the spherically supported delta distribution, where the distribution takes the value $+\infty$ on the dashed circle and is identically zero elsewhere. Right: with regularization, the support of the distribution is smoothed such that the distribution takes finite values and can be sampled on the underlying discrete grid.

To extend these 1-D ideas to multiple dimensions, we use the tensor product formulation of Peskin [16], which obeys the following consistency result.

Theorem 2 (consistency of multidimensional discrete deltas [19]). *Let Γ be a continuous and bounded surface, $g \in C(\mathbb{R}^d)$, and $\delta_{h_k} \in Q^q$ for $k = 1, \ldots, d$. Define the multidimensional function*

$$\delta_h(\Gamma, g, \boldsymbol{x}) \equiv \int_\Gamma \prod_{k=1}^{d} \delta_{h_k}(x_k - z_k) g(\boldsymbol{z})\, d\boldsymbol{z}, \tag{31}$$

where $\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$ and $\boldsymbol{z} = (z_1, \ldots, z_d) \in \Gamma$. Suppose $f \in C^r(\mathbb{R}^d)$. Then

$$\left(\prod_{k=1}^{d} h_k \right) \sum_{j \in \mathbb{Z}^d} \delta_h(\Gamma, g, \boldsymbol{x}_j) f(\boldsymbol{x}_j) - \int_\Gamma g(\boldsymbol{z}) f(\boldsymbol{z})\, d\boldsymbol{z} = \mathcal{O}(h^q), \tag{32}$$

where $\boldsymbol{x}_j = (j_1 h_1, \ldots, j_d h_d)$ is a Cartesian grid point and $h = \max_{k=1,\ldots,d} h_k$.

Intuitively, Theorem 2 gives a method to evaluate integrals of the form (26) by regularizing the singular surface via convolution with the multidimensional smoothing kernel δ_h. For example, with $\Gamma = \partial B_{\Delta t}$ and $g(\boldsymbol{z}) \equiv 1$, we see that

$$\delta_h(\partial B_{\Delta t}, 1, \boldsymbol{x}) = \delta(|\boldsymbol{x}| - \Delta t) * \left(\prod_{k=1}^{d} \delta_{h_k}(x_k - z_k) \right), \tag{33}$$

so Theorem 2 essentially permits accurate discrete convolution against $\delta(|\boldsymbol{x}| - \Delta t)$ by presmoothing the sphere with the kernel $\phi(\boldsymbol{x}) = \prod_{k=1}^{d} \delta_{h_k}(x_k)$ prior to sampling; see Figure 1. This is the key step permitting accurate discretization of the singular convolutional operators comprising our propagators.

3.2. ***Spherical quadrature.*** Discretization of the convolutional operators $G^{\Delta t}$ and $G_d^{\Delta t}$ necessitates the generation of samples of the function $\delta_h(\Gamma, g, \boldsymbol{x})$ in (31), where $\Gamma = \partial B_{\Delta t}$ and the weighting function g is defined by either $g(\boldsymbol{z}) \equiv 1$ or $g(\boldsymbol{z}) = z_i$,

depending on the kernel. In practice, we find that the integral in (31) has no simple analytical solution and therefore must be replaced with some form of quadrature scheme. We use the product Gaussian quadrature described in [3],

$$\begin{aligned}\delta_h(\partial B_{\Delta t}, g, \boldsymbol{x}) &= \int_{\partial B_{\Delta t}} \left(\prod_{k=1}^{d} \delta_{h_k}(x_k - z_k) g(\boldsymbol{z}) \right) d\boldsymbol{z}, \\ &\approx \frac{\pi(\Delta t)^2}{m} \sum_{j=1}^{2m} \sum_{i=1}^{m} w_i \left(\prod_{k=1}^{d} \delta_{h_k}(x_k - z_{ij,k}) g(\boldsymbol{z}_{ij}) \right), \end{aligned} \tag{34}$$

where $\boldsymbol{z}_{ij}$ has polar coordinates $(\Delta t, \theta_i, \phi_j)$ with $\cos\theta_i$ and w_i the Gauss–Legendre nodes and weights on $[-1, 1]$ and ϕ_j evenly spaced on $[0, 2\pi]$.

We note that, typically, such numerical quadratures use assumptions on the smoothness of the integrand to prove convergence whereas the smoothness of the integrand in (31) is dependent on the smoothness of δ_{h_k}. However, we are not interested in the intermediate error in evaluating $\delta_h(\Gamma, g, \boldsymbol{x})$ on a grid but rather the operator error of $\delta_h(\Gamma, g, \boldsymbol{x})$ as a discrete convolutional operator when applied to a smooth function f. Up to machine precision, the numerical quadrature and the discrete convolution commute, and we thus achieve accuracy from our assumptions on the smoothness of f rather than that of δ_h.

3.3. ***Final construction of operators.*** Given a spline approximation to the 1-D delta distribution obeying moment conditions up to order q, we use the results from the previous section to construct discretizations $G^{\Delta t,h}$ and $H^{\Delta t,h}$ of the convolutional kernels $G^{\Delta t}$ and $H^{\Delta t}$ for a specified time step Δt as follows. Defining $G^{\Delta t,h}$ via

$$G^{\Delta t,h}(\boldsymbol{x}_i) \equiv \frac{1}{4\pi\,\Delta t} \left(\prod_{k=1}^{d} h_k \right) \delta_h(\partial B_{\Delta t}, 1, \boldsymbol{x}_i), \tag{35}$$

we see from Theorem 2 that the approximation $G^{\Delta t} * f \approx G^{\Delta t,h} * f_h$ is pointwise accurate to order q, where the second convolution is understood as discrete convolution of $G^{\Delta t,h}$ with f sampled on a Cartesian grid. As discussed in Section 3.2, we use product Gaussian quadrature to evaluate the spherical integral necessary to construct the discrete delta distribution. The accuracy of this scheme thus necessarily depends on the number of quadrature nodes used, but this can be taken to be sufficiently high since $G^{\Delta t,h}$ need be constructed only once as a precomputation. In practice, we find that the number of quadrature nodes is not a limiting factor; see Section 5.1.

To construct the discrete kernel $H^{\Delta t,h}$, we note from (16) that we require a discrete approximation of the weighted kernels $G_d^{\Delta t} = z_d \delta(|\boldsymbol{z}| - \Delta t)$ for $d = 1, 2, 3$, which we construct similarly to the kernel in (35) by taking $g(\boldsymbol{z}) = z_d$. Then, approximating the spatial derivatives by precomposing the discrete kernels $G_d^{\Delta t,h}$

Initialize $p_h^{(0)}$ and $\boldsymbol{v}_h^{(0)}$
Compute $G^{\Delta t,h}$, and $H^{\Delta t,h}$ based on step size in time and space
`/* Begin time-stepping loop */`
for $n = 1, 2, \ldots$
`/* Update velocities */`
$v_{x,h}^{(n)} \leftarrow H^{\Delta t,h} * v_{x,h}^{(n-1)} + (G^{\Delta t,h} * \Delta_x) * p_h^{(n-1)}$
$v_{y,h}^{(n)} \leftarrow H^{\Delta t,h} * v_{y,h}^{(n-1)} + (G^{\Delta t,h} * \Delta_y) * p_h^{(n-1)}$
$v_{z,h}^{(n)} \leftarrow H^{\Delta t,h} * v_{z,h}^{(n-1)} + (G^{\Delta t,h} * \Delta_z) * p_h^{(n-1)}$
`/* Update pressure */`
$p_h^{(n)} \leftarrow H^{\Delta t,h} * p_h^{(n-1)} + (G^{\Delta t,h} * \Delta_x) * v_{x,h}^{(n-1)} + (G^{\Delta t,h} * \Delta_y) * v_{y,h}^{(n-1)} + (G^{\Delta t,h} * \Delta_z) * v_{z,h}^{(n-1)}$
end for

Algorithm 1. Applying the wave equation propagator.

with corresponding finite difference stencils Δ_{x_d}, we obtain $H^{\Delta t,h}$ as

$$H^{\Delta t,h} \equiv \frac{1}{\Delta t} G^{\Delta t,h} - \sum_{d=1}^{3} G_d^{\Delta t,h} * \Delta_{x_d} \tag{36}$$

using the equivalent expression (16). We choose central difference stencils accurate to order q for consistency. For example, if $q = 4$, then we use the typical fourth-order central difference [7]

$$(\Delta_{x_d} * f_h)(\boldsymbol{x}_i) = \frac{\frac{1}{12} f_h(\boldsymbol{x}_i - 2\boldsymbol{e}_d) - \frac{2}{3} f_h(\boldsymbol{x}_i - \boldsymbol{e}_d) + \frac{2}{3} f_h(\boldsymbol{x}_i + \boldsymbol{e}_d) - \frac{1}{12} f_h(\boldsymbol{x}_i + 2\boldsymbol{e}_d)}{h}, \tag{37}$$

where $\boldsymbol{e}_d$ is the d-th unit coordinate vector.

We note that, just as the continuous kernels $G^{\Delta t}$ and $H^{\Delta t}$ are compactly supported, so too are the discrete kernels with only slightly larger support size dependent on the exact smoothing splines and finite difference stencils used.

3.4. ***A typical time step.*** For source-free applications, the Maxwell and wave equation propagators can in theory be constructed for $\Delta t = t_{\text{final}}$ and applied as a one-step method. However, for many applications of interest, such as particle-in-cell (PIC) methods, it is necessary to evolve the solution only by a small time increment so that sources can be computed and incorporated. Here we describe the computation loop for time-stepping using the discrete kernels $G^{\Delta t,h}$ and $H^{\Delta t,h}$ described in Section 3.3. Consider first the wave equation propagator of (10). Then, the solution of the source-free problem (5) is obtained using the time-stepping process in Algorithm 1. We use $v_{x,h}$, $v_{y,h}$, and $v_{z,h}$ to refer to the discrete representations of components of $\boldsymbol{v}$ and Δ_x, Δ_y, and Δ_z to refer to finite difference operators in each coordinate direction as defined in Section 3.3. As mentioned, the discrete kernels $G^{\Delta t,h}$ and $H^{\Delta t,h}$ are compactly supported. This admits the use of Hockney's domain-doubling method [11] to evaluate all discrete convolutions efficiently.

/* Electric field and source are transverse throughout */
Initialize $\boldsymbol{E}_h^{(0)}$ and $\boldsymbol{B}_h^{(0)}$
Initialize Newton–Cotes quadrature weights $\{w_m\}_{m=0}^{M}$
Compute $G^{c\Delta s,h}$ and $H^{c\Delta s,h}$ based on step size in time and space
/* Begin time-stepping loop */
for $n = 1, 2, \ldots$
  /* Initialize for Newton-Cotes */
  $\boldsymbol{E}_h^{(n)} \leftarrow \boldsymbol{E}_h^{(n-1)}$
  $\boldsymbol{B}_h^{(n)} \leftarrow \boldsymbol{B}_h^{(n-1)}$
  for $m = 0, 1 \ldots, M-1$
    /* Add source term for node $t_{n,m} = (n-1)\Delta t + m\Delta s$ */
    $\boldsymbol{E}_h^{(n)} \leftarrow \boldsymbol{E}_h^{(n)} - w_m \boldsymbol{J}_h(t_{n,m})$
    /* Update electric field */
    $E_{x,h}^{(n)} \leftarrow H^{c\Delta s,h} * E_{x,h}^{(n)} + (G^{c\Delta s,h} * \Delta_y) * B_z^{(n)} - (G^{c\Delta s,h} * \Delta_z) * B_y^{(n)}$
    $E_{y,h}^{(n)} \leftarrow H^{c\Delta s,h} * E_{y,h}^{(n)} + (G^{c\Delta s,h} * \Delta_z) * B_x^{(n)} - (G^{c\Delta s,h} * \Delta_x) * B_z^{(n)}$
    $E_{z,h}^{(n)} \leftarrow H^{c\Delta s,h} * E_{z,h}^{(n)} + (G^{c\Delta s,h} * \Delta_x) * B_y^{(n)} - (G^{c\Delta s,h} * \Delta_y) * B_x^{(n)}$
    /* Update magnetic field */
    $B_{x,h}^{(n)} \leftarrow H^{c\Delta s,h} * B_{x,h}^{(n)} - (G^{c\Delta s,h} * \Delta_y) * E_{z,h}^{(n)} + (G^{c\Delta s,h} * \Delta_z) * E_{y,h}^{(n)}$
    $B_{y,h}^{(n)} \leftarrow H^{c\Delta s,h} * B_{y,h}^{(n)} - (G^{c\Delta s,h} * \Delta_z) * E_{x,h}^{(n)} + (G^{c\Delta s,h} * \Delta_x) * E_{z,h}^{(n)}$
    $B_{z,h}^{(n)} \leftarrow H^{c\Delta s,h} * B_{z,h}^{(n)} - (G^{c\Delta s,h} * \Delta_x) * E_{y,h}^{(n)} + (G^{c\Delta s,h} * \Delta_y) * E_{x,h}^{(n)}$
  end for
  /* Final Newton-Cotes node */
  $\boldsymbol{E}_h^{(n)} \leftarrow \boldsymbol{E}_h^{(n)} - w_M \boldsymbol{J}_h(t_{n,M})$
end for

Algorithm 2. Applying the Maxwell propagator.

Application of the transverse Maxwell propagator for (20) is analogous to Algorithm 1 in the source-free case. In the presence of sources, however, we must combine the basic flavor of the previous algorithm with a time-integration scheme for treating the integral in (3). To accomplish this, we use a closed Newton–Cotes method in time of the appropriate order with equispaced nodes $\{s_m = m\Delta s\}_{m=0}^{M}$ ($\Delta s = \Delta t/M$):

$$\int_0^{\Delta t} \mathcal{P}^{\Delta t - s}(\boldsymbol{f}_s)\, ds \approx \sum_{m=0}^{M} w_m \mathcal{P}^{\Delta t - s_m}(\boldsymbol{f}_{s_m}) = \sum_{m=0}^{M} w_m \mathcal{P}^{(M-m)\Delta s}(\boldsymbol{f}_{s_m}) \tag{38}$$

$$= \sum_{m=0}^{M} w_m \overbrace{\mathcal{P}^{\Delta s}(\mathcal{P}^{\Delta s}(\cdots \mathcal{P}^{\Delta s}(\boldsymbol{f}_{s_m})\cdots))}^{M-m \text{ times}}, \tag{39}$$

where, because the quadrature nodes are equispaced in time, we may make use of the fact that, analytically, application of the propagator $\mathcal{P}^{(M-m)\Delta s}$ corresponding to

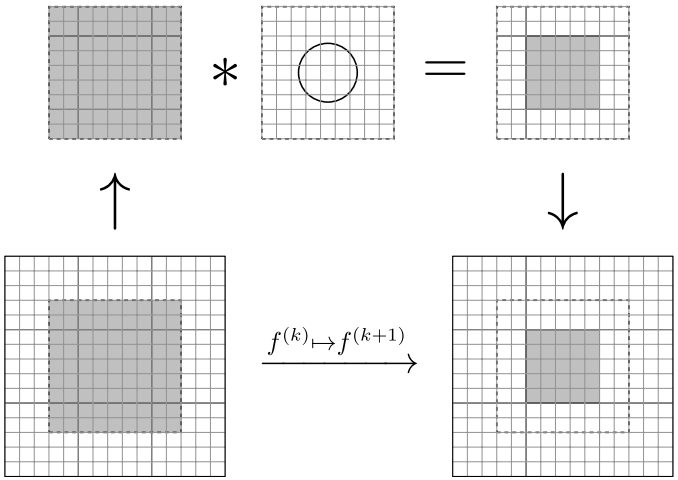


Figure 2. Suppose that the domain is decomposed across a 3×3 grid of processors such that the central processor owns the central block of unknowns as well as the halo region. Assuming the halo region is valid and sufficiently large, the local field values for time $n + 1$ can be calculated by convolving the current local-plus-halo field values with the propagator. In this figure, cells are shaded to show that the values of the corresponding unknowns are correct for the current time on the central processor.

advancing the solution in time by $(M - m)\Delta s$ is equivalent to $M - m$ successive applications of the single-step propagator $\mathcal{P}^{\Delta s}$, necessitating construction of only a single discrete propagator. This leads to Algorithm 2 for the transverse Maxwell propagator with transverse current source $\boldsymbol{J}$.

Because we are considering using the short-time propagators in a time-stepping loop, we have to consider the stability properties of the repeated application of these propagators. We discuss this in Section 4. While we do not discuss it in detail here, the number of nodes in the composite Newton–Cotes scheme described in (38) can also affect stability.

3.5. ***Parallelization.*** The use of a compactly supported convolutional kernel to regularize the 3-D delta distributions inherent in the propagators of (10) and (24) has the computational benefit of numerically preserving the locality inherent in the wave equation. In particular, in the same vein as Vay et al. [20], we can use standard domain decomposition to parallelize the time-stepping procedure as in Figure 2. This is in contrast to methods that use spectral expansions or global Fourier transforms to obtain high accuracy.

Parallelization of our scheme follows the traditional communication-computation loop of standard finite difference schemes:

(1) copy field values in halo region from neighboring processors, and then

(2) apply propagator to update local field values, invalidating values in halo region.

Note that the width of the halo region here has a minimum bound dictated by the size of the time step Δt since the size of the support of the spherical delta distributions is dependent on how far in time the fields are to be advanced. For example, in the presence of sources as in Algorithm 2, the size of the halo region should be such that communication is only necessary after M applications of the discrete propagator constructed with step size $\Delta s = \Delta t/M$. Thus, Δt is chosen based on the desired size of the halo region and Δs, which determines the CFL number, is chosen based on the necessary resolution to resolve variation in the source.

4. Stability analysis

Letting $\mathcal{P}_M^{\Delta s,h}$ denote the discrete Maxwell propagator described by Algorithm 2, we see that the evolution of the discretized electromagnetic fields is given by

$$\begin{bmatrix} \boldsymbol{E}_h^{(n)} \\ \boldsymbol{B}_h^{(n)} \end{bmatrix} = \mathcal{P}_M^{\Delta s,h}\left(\begin{bmatrix} \boldsymbol{E}_h^{(n-1)} \\ \boldsymbol{B}_h^{(n-1)} \end{bmatrix}\right).$$

Taking a Fourier transform, we obtain the Fourier-space relation

$$\begin{bmatrix} \widetilde{\boldsymbol{E}}_h^{(n)} \\ \widetilde{\boldsymbol{B}}_h^{(n)} \end{bmatrix} = \widetilde{\mathcal{P}}_M^{\Delta s,h} \begin{bmatrix} \widetilde{\boldsymbol{E}}_h^{(n-1)} \\ \widetilde{\boldsymbol{B}}_h^{(n-1)} \end{bmatrix},$$

where now $\widetilde{\mathcal{P}}_M^{\Delta s,h}$ is a matrix in $\boldsymbol{k}$ space (as opposed to a sum of convolutional operators in physical space). For a typical von Neumann analysis of ℓ_2 stability, we define $\rho(\boldsymbol{k})$ to be the spectral radius of $\widetilde{\mathcal{P}}_M^{\Delta s,h}$ and show that the necessary condition (see, e.g., [17])

$$\rho(\boldsymbol{k}) \leq 1 + \mathcal{O}(\Delta s)$$

holds for all $\boldsymbol{k}$. For analysis purposes, we assume that the integrals used to construct the necessary regularized, spherically supported delta distributions are computed exactly, i.e., without the use of the quadrature described in Section 3.2. Then, direct computation shows that $\widetilde{\mathcal{P}}_M^{\Delta s,h}(\boldsymbol{k})$ is given by

$$\widetilde{\mathcal{P}}_M^{\Delta s,h} = \begin{bmatrix} \widetilde{H}^{c\Delta s,h} & 0 & 0 & 0 & -\widetilde{G}^{c\Delta s,h}\tilde{\Delta}_z & \widetilde{G}^{c\Delta s,h}\tilde{\Delta}_y \\ 0 & \widetilde{H}^{c\Delta s,h} & 0 & \widetilde{G}^{c\Delta s,h}\tilde{\Delta}_z & 0 & -\widetilde{G}^{c\Delta s,h}\tilde{\Delta}_x \\ 0 & 0 & \widetilde{H}^{c\Delta s,h} & -\widetilde{G}^{c\Delta s,h}\tilde{\Delta}_y & \widetilde{G}^{c\Delta s,h}\tilde{\Delta}_x & 0 \\ 0 & \widetilde{G}^{c\Delta s,h}\tilde{\Delta}_z & -\widetilde{G}^{c\Delta s,h}\tilde{\Delta}_y & \widetilde{H}^{c\Delta s,h} & 0 & 0 \\ -\widetilde{G}^{c\Delta s,h}\tilde{\Delta}_z & 0 & \widetilde{G}^{c\Delta s,h}\tilde{\Delta}_x & 0 & \widetilde{H}^{c\Delta s,h} & 0 \\ \widetilde{G}^{c\Delta s,h}\tilde{\Delta}_y & -\widetilde{G}^{c\Delta s,h}\tilde{\Delta}_x & 0 & 0 & 0 & \widetilde{H}^{c\Delta s,h} \end{bmatrix},$$

where the ordering of the blocks corresponds to the vector $\left[\widetilde{E}_x\ \widetilde{E}_y\ \widetilde{E}_z\ \widetilde{B}_x\ \widetilde{B}_y\ \widetilde{B}_z\right]^T$ and the blocks of $\widetilde{\mathcal{P}}_M^{\Delta s,h}$ are functions of the Fourier variable $\boldsymbol{k}$ corresponding to the transforms of the discrete operators from Section 3.3.

Lemma 3. *The eigenvalues of* $\widetilde{\mathcal{P}}_M^{\Delta s,h}$ *as a function of* $\boldsymbol{k}$ *are given by*

$$\lambda_{1,2}(\boldsymbol{k}) = \widetilde{H}^{c\Delta s,h}(\boldsymbol{k}) \pm \widetilde{G}^{c\Delta s,h}(\boldsymbol{k})\sqrt{\tilde{\Delta}_x^2(\boldsymbol{k}) + \tilde{\Delta}_y^2(\boldsymbol{k}) + \tilde{\Delta}_z^2(\boldsymbol{k})},$$
$$\lambda_3(\boldsymbol{k}) = \widetilde{H}^{c\Delta s,h}(\boldsymbol{k}),$$

each of which appears with multiplicity 2.

Proof. This follows from direct computation via block linear algebra. □

To obtain explicit expressions for the eigenvalues in Lemma 3, we represent the sampling operator using:

Definition 4. Given the spatial step sizes h_1, h_2, and h_3, the 3-D Dirac comb Ш_h is defined on $\mathbb{R}^3$ as

$$\text{Ш}_h(\boldsymbol{x}) \equiv \sum_{\boldsymbol{l}\in\mathbb{Z}^3}\left(\prod_{d=1}^{3}\delta(x_d - l_d h_d)\right).$$

In other words, Ш_h is a regular 3-D lattice of delta distributions.

For the methods described in this paper, the regularized 1-D delta distribution used to construct the regularized 3-D delta distributions in $G^{c\Delta s,h}$ and $H^{c\Delta s,h}$ is given by shifting and scaling of some fundamental kernel W, i.e.,

$$\delta_{h_d}(x) \equiv \frac{1}{h_d} W(x/h_d), \tag{40}$$

which we combine with the sampling operator to explicitly write the Fourier transforms $\widetilde{G}^{c\Delta s,h}$ and $\widetilde{H}^{c\Delta s,h}$ as

$$\widetilde{H}^{c\Delta s,h}(\boldsymbol{k}) = \int_{\mathbb{R}^3} \text{Ш}_{2\pi/h}(\boldsymbol{k}-\boldsymbol{k}')\left[\left(\prod_{d=1}^{3}\widetilde{W}(k_d' h_d)\right)\widetilde{P}_1(\boldsymbol{k}')\right]d\boldsymbol{k}', \tag{41}$$

$$\widetilde{G}^{c\Delta s,h}(\boldsymbol{k}) = \int_{\mathbb{R}^3} \text{Ш}_{2\pi/h}(\boldsymbol{k}-\boldsymbol{k}')\left[\left(\prod_{d=1}^{3}\widetilde{W}(k_d' h_d)\right)\widetilde{P}_2(\boldsymbol{k}')\right]d\boldsymbol{k}', \tag{42}$$

where the functions $\widetilde{P}_1$ and $\widetilde{P}_2$ are defined in Fourier space according to

$$\widetilde{P}_1(\boldsymbol{k}) \equiv \frac{\sin c|\boldsymbol{k}|\Delta s}{c|\boldsymbol{k}|\Delta s} - i\sum_{d=1}^{3}\frac{\partial}{\partial k_d}\left[\frac{\sin c|\boldsymbol{k}|\Delta s}{c|\boldsymbol{k}|\Delta s}\right]\tilde{\Delta}_{x_d}(k_d),$$
$$\widetilde{P}_2(\boldsymbol{k}) \equiv \frac{\sin c|\boldsymbol{k}|\Delta s}{|\boldsymbol{k}|}.$$

We are now ready to state and prove the fundamental stability result for the discrete Maxwell propagator.

Theorem 5. *In Algorithm 2, assume that the regularized, spherically supported delta distributions are computed exactly, i.e., without the use of quadrature. Suppose further that the 1-D delta distributions are constructed using a fundamental kernel W as in (40) satisfying the following properties*:

(1) $\widetilde{W}(k)$ *is real and nonnegative for* $k \in \mathbb{R}$.

(2) $W(j) = 0$ *for* $j \in \mathbb{Z}$ *except* $W(0) = 1$; *i.e., W behaves like the Kronecker delta on the lattice points.*

In addition, suppose that for each d the finite difference stencil Δ_{x_d} has real coefficients and odd symmetry and has spectrum bounded for $|k_d h_d| \in (0, \pi]$ as

$$0 \leq \frac{\tilde{\Delta}_{x_d}(k_d)}{ik_d} \leq 1.$$

Furthermore, define the quantity

$$\widetilde{R}(\boldsymbol{k}) \equiv \sum_{d=1}^{3} \frac{-ik_d \tilde{\Delta}_{x_d}(k_d)}{|\boldsymbol{k}|^2},$$

and assume that there exists a bound $B < 1$ such that $|\widetilde{R}(\boldsymbol{k})| \leq B < 1$ for all $\boldsymbol{k}$ with any $|k_d| \geq \pi / h_d$. Then the time-stepping scheme satisfies the von Neumann condition for

$$\sigma \equiv \frac{c\Delta s}{h} \geq \frac{1+B}{\pi(1-B)},$$

where $h = \max_d h_d$.

Proof. Based on the assumption that Δ_{x_d} is a real and odd finite difference stencil, it has a purely imaginary Fourier transform. Using Lemma 3, we deduce that $\mathcal{P}_M^{\Delta s,h}(\boldsymbol{k})$ has a spectral radius ρ given by

$$\rho(\boldsymbol{k}) = \left| \widetilde{H}^{c\Delta s,h}(\boldsymbol{k}) + \widetilde{G}^{c\Delta s,h}(\boldsymbol{k}) \sqrt{\tilde{\Delta}_x^2(\boldsymbol{k}) + \tilde{\Delta}_y^2(\boldsymbol{k}) + \tilde{\Delta}_z^2(\boldsymbol{k})} \right|.$$

Using the triangle inequality and plugging in expressions (41) and (42) yields

$$\rho(\boldsymbol{k}) \leq \int_{\mathbb{R}^3} \text{Ш}_{2\pi/h}(\boldsymbol{k} - \boldsymbol{k}') \left[\left(\prod_{d=1}^{3} \widetilde{W}(k'_d h_d) \right) \cdot |\tilde{S}(\boldsymbol{k}')| \right] d\boldsymbol{k}'$$

with the quantity $\tilde{S}(\boldsymbol{k})$ defined according to

$$\tilde{S}(\boldsymbol{k}) \equiv \widetilde{P}_1(\boldsymbol{k}) + \widetilde{P}_2(\boldsymbol{k}) \sqrt{\tilde{\Delta}_x^2(\boldsymbol{k}) + \tilde{\Delta}_y^2(\boldsymbol{k}) + \tilde{\Delta}_z^2(\boldsymbol{k})}. \tag{43}$$

We will show that $\tilde{S}$ has magnitude bounded by 1 for all k and use this to show that ρ is bounded by 1 for all k.

Using some calculus and our definition of the quantity $\widetilde{R}$, we write the term $\widetilde{P}_1$ as

$$\widetilde{P}_1(\boldsymbol{k}) = \frac{\sin c|\boldsymbol{k}|\Delta s}{c|\boldsymbol{k}|\Delta s}(1 - \widetilde{R}(\boldsymbol{k})) + \widetilde{R}(\boldsymbol{k}) \cos c|\boldsymbol{k}|\Delta s,$$

which we note is purely real. Furthermore, we see that the second term in (43) is purely imaginary. We proceed by breaking the argument across two separate cases.

First, assume that $\boldsymbol{k}$ is such that $|k_d h_d| \in [0, \pi]$ for all d. Then by assumption, $\widetilde{R}(\boldsymbol{k}) \in [0, 1]$. We use convexity to see that the real part of $\tilde{S}(\boldsymbol{k})$ squared is bounded according to

$$\begin{aligned}\mathrm{Re}[\tilde{S}(\boldsymbol{k})]^2 &= \left(\frac{\sin c|\boldsymbol{k}|\Delta s}{c|\boldsymbol{k}|\Delta s}(1 - \widetilde{R}(\boldsymbol{k})) + \widetilde{R}(\boldsymbol{k}) \cos c|\boldsymbol{k}|\Delta s\right)^2 \\ &\leq \left(\frac{\sin c|\boldsymbol{k}|\Delta s}{c|\boldsymbol{k}|\Delta s}\right)^2 (1 - \widetilde{R}(\boldsymbol{k})) + \widetilde{R}(\boldsymbol{k})(\cos c|\boldsymbol{k}|\Delta s)^2.\end{aligned}$$

On the other hand, the squared imaginary part of $\tilde{S}(\boldsymbol{k})$ is bounded according to

$$\mathrm{Im}[\tilde{S}(\boldsymbol{k})]^2 = \sin^2 c|\boldsymbol{k}|\Delta s\left(\frac{|\tilde{\Delta}_x^2(\boldsymbol{k}) + \tilde{\Delta}_y^2(\boldsymbol{k}) + \tilde{\Delta}_z^2(\boldsymbol{k})|}{|\boldsymbol{k}|^2}\right) \leq \widetilde{R}(\boldsymbol{k}) \sin^2 c|\boldsymbol{k}|\Delta s.$$

Combining these two bounds and using the fact that $\cos^2(x) + \sin^2(x) = 1$, it is simple to show that $|\tilde{S}(\boldsymbol{k})|^2 \leq 1$ as desired.

Next, suppose that $|k_d| > \pi/h_d$ for some d. By assumption, $|\widetilde{R}| \leq B$ and further it is evident that $|\boldsymbol{k}| \geq \pi/h_d$. In this case, we compute that the real part of $\tilde{S}(\boldsymbol{k})$ is given by

$$\begin{aligned}\mathrm{Re}[\tilde{S}(\boldsymbol{k})]^2 = \left(\frac{\sin c|\boldsymbol{k}|\Delta s}{c|\boldsymbol{k}|\Delta s}\right)^2 &(1 - \widetilde{R}(\boldsymbol{k}))^2 + \widetilde{R}^2(\boldsymbol{k}) \cos^2 c|\boldsymbol{k}|\Delta s \\ &+ 2(1 - \widetilde{R}(\boldsymbol{k}))\widetilde{R}(\boldsymbol{k}) \cos c|\boldsymbol{k}|\Delta s \frac{\sin c|\boldsymbol{k}|\Delta s}{c|\boldsymbol{k}|\Delta s}.\end{aligned}$$

Using the identity $2\cos(x)\sin(x) = \sin(2x)$ and the bound B from our assumptions, we see

$$\mathrm{Re}[\tilde{S}(\boldsymbol{k})]^2 \leq \left(\frac{\sin c|\boldsymbol{k}|\Delta s}{c|\boldsymbol{k}|\Delta s}\right)^2 (1 + B)^2 + B \cos^2 c|\boldsymbol{k}|\Delta s + B(1 + B)\left|\frac{\sin 2c|\boldsymbol{k}|\Delta s}{c|\boldsymbol{k}|\Delta s}\right|.$$

We bound the sines by 1 and the magnitudes of $\boldsymbol{k}$ by π/h_d to obtain

$$\begin{aligned}\mathrm{Re}[\tilde{S}(\boldsymbol{k})]^2 &\leq \left(\frac{1}{c\pi/h_d\Delta s}\right)^2 (1 + B)^2 + B \cos^2 c|\boldsymbol{k}|\Delta s + B(1 + B)\left|\frac{1}{c\pi/h_d\Delta s}\right| \\ &\leq \frac{1}{\sigma^2\pi^2}(1 + B)^2 + B \cos^2 c|\boldsymbol{k}|\Delta s + B(1 + B)\frac{1}{\sigma\pi}.\end{aligned}$$

The imaginary part of $\tilde{S}(\boldsymbol{k})$ is bounded trivially as before since

$$\mathrm{Im}[\tilde{S}(\boldsymbol{k})]^2 \leq |\widetilde{R}(\boldsymbol{k})| \sin^2 c|\boldsymbol{k}|\Delta s \leq B \sin^2 c|\boldsymbol{k}|\Delta s.$$

Combining these two bounds and assuming $\sigma \geq (1+B)/(\pi(1-B))$ again gives that $|\tilde{S}(\boldsymbol{k})|^2 \leq 1$.

Now we have shown that $|\tilde{S}(\boldsymbol{k})| \leq 1$ for all $\boldsymbol{k}$, from which it immediately follows that $\rho(\boldsymbol{k}) \leq 1$ since

$$\begin{aligned}
\rho(\boldsymbol{k}) &\leq \int_{\mathbb{R}^3} \mathrm{III}_{2\pi/h}(\boldsymbol{k}-\boldsymbol{k}') \left[\left(\prod_{d=1}^{3} \widetilde{W}(k'_d h_d) \right) \cdot |\tilde{S}(\boldsymbol{k}')| \right] d\boldsymbol{k}' \\
&\leq \int_{\mathbb{R}^3} \mathrm{III}_{2\pi/h}(\boldsymbol{k}-\boldsymbol{k}') * \left(\prod_{d=1}^{3} \widetilde{W}(k'_d h_d) \right) d\boldsymbol{k}' \\
&= \prod_{d=1}^{3} \left(\sum_{l\in\mathbb{Z}} \widetilde{W}(k_d h_d + 2\pi l) \right) = 1,
\end{aligned}$$

where the last equality comes from the fact that by Poisson summation we have that $\sum_{l\in\mathbb{Z}} \widetilde{W}(kh+2\pi l)$ is the discrete-space Fourier transform of W evaluated at kh, which is unity everywhere by the fact that W behaves like the Kronecker delta on the lattice points. This concludes the proof. □

The main result of Theorem 5 is somewhat odd in the sense that we have shown the von Neumann condition holds for all σ above a certain minimum value whereas in general we expect stability results to give an upper bound on σ. We believe this to simply be an artifact of the proof technique employed.

To extend Theorem 5 to the wave equation propagator, we note that for this new propagator the eigenvalues in Fourier space are given by

$$\begin{aligned}
\lambda^W_{1,2}(\boldsymbol{k}) &= \widetilde{H}^{\Delta s,h}(\boldsymbol{k}) \pm \widetilde{G}^{\Delta s,h}(\boldsymbol{k}) \sqrt{\tilde{\Delta}^2_x(\boldsymbol{k}) + \tilde{\Delta}^2_y(\boldsymbol{k}) + \tilde{\Delta}^2_z(\boldsymbol{k})}, \\
\lambda^W_3(\boldsymbol{k}) &= \widetilde{H}^{\Delta s,h}(\boldsymbol{k})
\end{aligned}$$

with varying multiplicity. Analysis analogous to that in the proof of Theorem 5 proceeds in a similar fashion by taking $c = 1$. For the source-free case, it is not necessary to use a quadrature scheme and thus one can take $\Delta t = \Delta s$.

We must caution that, in proving Theorem 5, we have assumed that the spherical integrals are performed exactly, which in practice is not possible. It is unclear whether the use of a quadrature scheme such as described in Section 3.2 might lead to unstable modes. Our numerical experiments, however, show no evidence of instability.

5. Numerical experiments

We have performed convergence studies using two tests: one where the exact solution is known, where we can compute the true absolute error, and another where the exact solution is unknown, where we estimate the convergence rate using Richardson error estimation. We also present numerical test results investigating any dependence the solution has on domain decomposition and number of points in the spherical quadrature of (34). Lastly, we present timing results demonstrating the weak scaling of our solver.

The following notation is used throughout the section:

N: number of grid points per spatial dimension,

h: grid spacing $(1/N)$,

σ: CFL number $(c\Delta t/h)$,

N_θ: number of points in the θ direction for the spherical quadrature,

N_t: number of time steps.

All tests are performed in a unit cube with N^3 points and $N_\theta = 16$ unless otherwise specified. All tests presented in this section were performed on the Edison machine at the NERSC facility.

5.1. ***Results.*** We implemented our method in C++ using the Chombo library [1]. Our implementation uses a sixth-order central difference stencil for the spatial derivatives, a sixth-order interpolation formula for the discrete delta distribution, and the sixth-order Boole's rule for the source integration. All convolutions are performed via Hockney's method with simple domain doubling using the FFTW library [8]. We note that it is in fact not necessary to fully double the domain to perform the convolutions for our solver — rather, we extend the domain with a number of grid cells equal to the support of the discrete propagator, which is usually much smaller than the local domain on each processor.

Plane wave. We begin by testing our code with no source ($\boldsymbol{J} = \mathbf{0}$ and $\rho = 0$) with periodic boundary conditions and initial conditions of the form

$$\begin{aligned} E_x(x, y, z) &= B_x(x, y, z) = 0, \\ E_y(x, y, z) &= B_z(x, y, z) = E_{y0} \sin 2\pi x, \\ E_z(x, y, z) &= E_{z0} \sin 2\pi x, \\ B_y(x, y, z) &= -E_z(x, y, z) \end{aligned}$$

in the domain $\Omega = [0, 1]^3$ m with $t_{\text{final}} = \frac{25}{8c}$ s. Solving Maxwell's equations with these initial conditions yields a set of plane waves propagating in the x direction with velocity c. Figure 3 shows the absolute error for $\sigma = [1, 10, 100]$ for the sixth-order

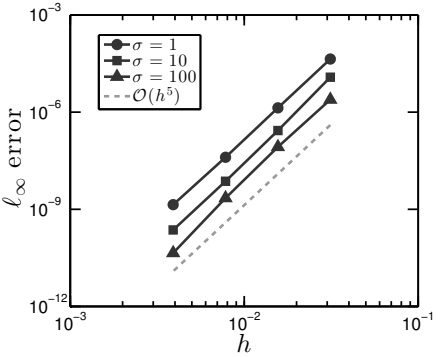


Figure 3. Max-norm error for the plane wave problem for $\sigma = [1, 10, 100]$ with $N = [32, 64, 128, 256]$ with $\sigma N_t = [100, 200, 400, 800]$, which corresponds to $t_{\text{final}} = \frac{25}{8c}s$. The error scales as $h^{4.99}$ for $\sigma = 1$, $h^{5.22}$ for $\sigma = 10$, and $h^{5.28}$ for $\sigma = 100$. The $\sigma = 100$ problem uses $N_\theta = 128$, and the others use $N_\theta = 16$.

solver with $E_{y0} = E_{z0} = 1$. As expected with constant-σ tests, the results yield fifth-order convergence for the absolute error; i.e., we lose one order of accuracy since the number of time steps is inversely proportional to the spatial step size.

Divergence-free current source. For the second test, we begin with zero initial condition and zero charge density but with a divergence-free current density [5] of the form

$$J_x(x, y, z) = -\frac{(y - y_0)}{r} \sin\left(\frac{\pi r}{2a}\right) \cos^{10}\left(\frac{\pi r}{2a}\right) \cos^{11}\left(\frac{\pi (z - z_0)}{d}\right) \sin(2\pi \nu t),$$

$$J_y(x, y, z) = \frac{x - x_0}{r} \sin\left(\frac{\pi r}{2a}\right) \cos^{10}\left(\frac{\pi r}{2a}\right) \cos^{11}\left(\frac{\pi (z - z_0)}{d}\right) \sin(2\pi \nu t),$$

$$J_z(x, y, z) = 0,$$

where $r = r(x, y) \equiv \sqrt{(x - x_0)^2 + (y - y_0)^2}$. With this source, we solve Maxwell's equations in a unit cube with open boundary conditions and parameters $a = 0.25$ m, $d = x_0 = y_0 = z_0 = 0.5$ m, and $\nu = 149\,896\,229\ \text{s}^{-1}$ in the domain $\Omega = [0, 1]^3$ m to $t_{\text{final}} = \frac{5}{32c}$ s. This frequency was chosen to match the low-frequency test of Chilton [5]. For this problem, the z component of the electric field is $E_z = 0$ for all time. Table 1 shows the Richardson error estimate for test problems with $\sigma = [\frac{1}{2}, 1, 10]$ using the sixth-order solver. For $\sigma = \frac{1}{2}$ and $\sigma = 1$, the error estimate for the nonzero components of the EM fields show fifth-order convergence as expected and for E_z it shows sixth-order convergence. For $\sigma = 10$, we see unexpected higher-order convergence for the error, which requires more investigation. However, as we see in Figure 4, the solution differences between corresponding points as the spatial resolution varies seem to indicate that our test cases sit within the asymptotic regime (a necessary condition for the validity of Richardson error estimates).

Component	$\sigma = 0.5$			$\sigma = 1$			$\sigma = 10$		
	ℓ_∞	ℓ_1	ℓ_2	ℓ_∞	ℓ_1	ℓ_2	ℓ_∞	ℓ_1	ℓ_2
E_x	4.96	4.99	4.99	4.97	4.99	5.00	8.12	7.46	7.87
E_y	4.96	4.99	4.99	4.97	4.99	5.00	8.12	7.46	7.87
E_z	5.88	5.89	5.90	5.82	5.85	5.84	4.89	5.22	5.05
B_x	5.12	5.12	5.15	5.14	5.14	5.17	6.63	7.03	6.96
B_y	5.12	5.12	5.15	5.14	5.14	5.17	6.63	7.03	6.96
B_z	5.07	5.09	5.11	5.08	5.10	5.13	6.53	7.03	6.95

Table 1. Richardson error estimate of asymptotic rate using ℓ_∞, ℓ_1, and ℓ_2 norms for the specified current test $\sigma = [\frac{1}{2}, 1, 10]$ with $N = [129, 257, 513]$ and $\sigma N_t = [20, 40, 80]$, respectively, corresponding to $t_{\text{final}} = \frac{5}{32c}$ s.

For the Richardson error estimation, we solve the problem to the same final time with increasingly finer discretizations h, $h/2$, and $h/4$ and let u_i denote the solution corresponding to the grid with spacing i. We then sample the solution on the $h/4$ grid onto the $h/2$ grid and the solution on the $h/2$ grid onto the h grid. Letting $\mathcal{S}_i$ denote the sample operator that transfers a solution on a grid with spacing $i/2$ to a

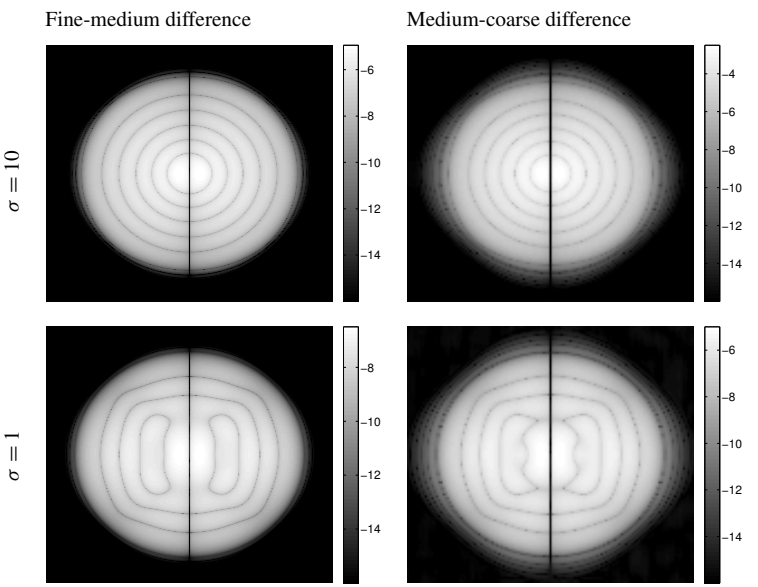


Figure 4. Visualizations of the logarithm (base 10) of the absolute difference between the solution on grids of different sizes, as used to obtain the antenna problem Richardson error estimates in Table 1. Our Richardson estimates use three grids at sizes $N = 513$ (fine), $N = 129$ (medium), and $N = 65$ (coarse). For example, to obtain the fine-medium difference, we subsample the values of E_x on the fine grid and measure the pointwise difference between the values at the corresponding locations on the medium grid. The figures here show a cross-sectional view of these quantities at the plane $z = 0.5$ m.

N_{patch}	N_θ	ℓ_∞ error	N_{patch}	N_θ	ℓ_∞ error
7	16	2.389×10^{-11}	9	32	6.343×10^{-12}
8	16	2.389×10^{-11}	9	64	3.543×10^{-11}
9	16	2.387×10^{-11}	9	128	4.175×10^{-11}
10	16	2.391×10^{-11}	9	256	1.031×10^{-11}

Table 2. Absolute ℓ_∞ error for the plane wave problem with $\sigma = 1$, $N = 513$, and $N_t = 160$, corresponding to $t_{\text{final}} = \frac{5}{16c}s$. The domain is decomposed into N_{patch}^3 number of subdomains.

grid with spacing i, the ℓ_p norm error rate estimate is given by

$$q = \frac{\log \left| u_{h/2} - \mathcal{S}_{h/2}(u_{h/4}) \right|_p - \log \left| u_h - \mathcal{S}_h(u_{h/2}) \right|_p}{\log \frac{1}{2}}. \tag{44}$$

For parallelization via domain decomposition, we break the domain into N_{patch}^3 subdomains and then solve the problem in parallel with a number of processors $N_{\text{proc}} = N_{\text{patch}}^3$. The error in our algorithm should not depend on N_{patch}, which we confirmed by solving the plane wave problem with fixed $\sigma = 1$, $N = 513$, and $N_t = 160$ and varying N_{patch}. The absolute ℓ_∞ error results can be seen in the left part of Table 2. As expected, the error shows no significant dependence on the subdomain sizes. Further, for this same plane wave problem, we investigated the dependance of the error on the number of discretization points used for the spherical quadrature, N_θ, which shows a slight decreasing trend as expected; see the right half of Table 2.

Table 3 shows weak scaling results of our algorithm in parallel applied to the prescribed current-source problem with CFL parameter $\sigma = \frac{1}{2}$. As shown by the normalized τ factor, our solver exhibits reasonable scaling once communication is introduced in the problem while it is lower for the single-processor case where no communication is necessary. Note that, while the number of processors scales with

N	N_t	N_{proc}	t_{solve} (s)	t_{quad}(s)	τ
65	20	1	613.65	0.02	1.11×10^{-4}
129	40	8	1471.29	0.03	1.37×10^{-4}
257	80	64	2791.86	0.03	1.32×10^{-4}
513	160	512	5830.04	0.03	1.38×10^{-4}

Table 3. Timing data for the prescribed current problem with $\sigma = \frac{1}{2}$ with the domain subdivided into $N_{\text{patch}}^3 = N_{\text{proc}}$ total subdomains solved on N_{proc} processors. The time spent in the solver, t_{solve}, and time spent doing the spherical quadrature, t_{quad}, are taken from the timing data of a single processor. The factor $\tau = t_{\text{solve}} N_{\text{proc}}/(N^3 N_t)$ is a normalized measure of time spent in the solver. In the perfect scaling case, τ would remain a constant.

the total number of points in the spatial discretization, the number of time steps increases by a factor of 2 between subsequent rows of the table, which is reflected in the way t_{solve} roughly doubles between rows.

6. Conclusion

Our numerical results demonstrate that we attain the desired order of accuracy through both a simple plane wave example (where the true solution is known) and a more complicated example with a time-dependent source (where we employ Richardson error estimates). The major advantage to this method is that it is easily parallelizable. This method does not have a CFL condition based on the wave speed so that the communication cost relative to an explicit method of same accuracy is much lower. The lack of a time step constraint is also a significant cost reduction when the field solver is used in PIC code. In addition, since the method is based on convolutions with compactly supported kernels, the Hockney algorithm is used on small patches of the domain; thus, it is well suited for multicore architectures with deep memory hierarchies. Our implementation has shown good weak scaling via parallelization in space. Our method is also computationally cheaper than explicit methods of the same accuracy for large CFL or large number of grid points. Suppose we would like to advance the solution by a time T using one time step of the present method. If we define $\sigma = \lceil cT/h \rceil$, then σ is the number of ghost points required in each direction per patch. The amount of work it takes to solve the problem per patch would then be the cost of the FFT and multiplication for Hockney, and therefore, it is $\mathcal{O}((N+\sigma)^3 \log(N+\sigma))$. On the other hand, for an explicit method, the CFL stability condition on the time step implies that the number of time steps required to advance the solution to time T is $\mathcal{O}(\sigma)$, and the total work required per patch would be $\mathcal{O}(N^3\sigma)$. Thus, we expect a decrease in the time to solution of the present method relative to an explicit method to be $\mathcal{O}(\log(N+\sigma)(1/N + 1/\sigma))$. If the execution time is dominated by the time to read and write the field data to and from cache, then the present algorithm is performing $\mathcal{O}(1/\sigma)$ as many such communication steps as an explicit method, and the time to solution is reduced by that factor.

In essence, the method presented in this paper solves the free-space Maxwell equations by assuming that the fields have been separated into local and nonlocal parts via Helmholtz decomposition and solving the local portion in parallel by constructing a compactly supported discrete convolutional kernel via

(1) finding an explicit analytic form for the Maxwell propagator,

(2) regularizing the singularities with convolutional smoothing kernels,

(3) replacing spatial derivatives with finite difference stencils, and

(4) sampling the result on a Cartesian grid.

Because the resulting discretized propagator takes the form of discrete convolution against a compactly supported kernel, a regular decomposition of the domain across processors with a halo region whose size is driven by the support of the propagator (i.e., the size of Δt and the order of discretization accuracy) admits exact application of the discrete propagator in parallel; i.e., the error is independent of the domain decomposition. Furthermore, by appropriate choice of the smoothing kernel and finite difference stencils, our method can attain an arbitrarily high order of accuracy. We view the rigorous error analysis and demonstration of accuracy as a strength of our paper and method. The method of [20] employs a similar idea for parallelization, using linearity and finite propagation speed to justify domain decomposition in the solution of Maxwell's equations. In fact, the continuous propagator for our method (24) is the same as the one used in [20]. However, the method the authors present advances all fields in Fourier rather than physical space, which they approximate in parallel by taking local FFTs on each subdomain. While they assert spectral accuracy of their method, they do not provide any analysis of he method, nor convergence results that would support such an assertion. The authors do note that the finite number of modes used in this representation leads qualitatively to small nonlocal errors, of which they defer analysis for later work.

It is important to keep in mind that the approach we show here assumes that the electric field has been decomposed everywhere via the Helmholtz decomposition and that the divergence-free component is to be treated by other methods. As such, future work will focus on coupling of our method with a fast and accurate method for Poisson's equation in order to compute the solution to (22) and/or (23) in the specific application to PIC methods for the Maxwell–Vlasov equations. While it may appear that, by looking only at the case of $\boldsymbol{J}_L \equiv 0$, we are ignoring a large computational cost, the trend in PIC methods for charged systems is towards a large number of particles per cell, i.e., hundreds or thousands, for the purpose of minimizing numerical noise. For electrostatic problems, the need for such large numbers of particles per cell is indicated by the convergence theory for PIC methods [21]. In this regime, the field solve constitutes a small fraction of the cost, even with a Poisson solve included. However, for classical explicit time-stepping methods, the time step for the overall calculation is constrained by the CFL condition for the Maxwell solve so that the ability to use larger time steps provides an additional tool to improve overall performance. Another area we will investigate is the extension of this method to locally refined grids, using an approach analogous to that in [14] for Poisson's equation. Since our key motivation for this method is to use it in a PIC method to simulate free plasmas where the EM waves radiate out with open boundary conditions, we are not concerned with dealing with other boundary conditions with this method. However, incorporating boundary conditions in integral evolution methods for the wave equation has been examined [13].

Finally, in the present method, we use the special structure of Maxwell's equations and the wave equation in 3-D to compute an analytic form for the propagator with support on the surface of the sphere corresponding to the wave front. While the propagator in 3-D involves integrating on the surface of a sphere, the propagator does not take this form in general. For instance, the 2-D wave equation propagator requires integration on a disc [22]. Therefore, it is not obvious if we could extend our idea of regularization to other dimensions. Nonetheless, there are a number of ways in which we could attempt to generalize this approach. One approach would be to construct a discrete propagator directly, using iterates of an explicit method applied to discrete-delta-function initial data. This would be done at most once per time step on a small patch and then applied multiple times as a discrete convolution kernel, as above. Another approach would be to use geometrical optics [12] as a starting point for constructing a sufficiently accurate approximate propagator to represent the stiff wave propagation. The first problem to which to apply either of these approaches would be the wave equation in 2-D, followed by the linearized Euler equations in the low-Mach number limit or linearized MHD in the low-Alfvén number limit.

Appendix: Constructing compactly supported delta approximations

To derive a function δ_h satisfying the discrete moment conditions, we first define the unscaled approximant W such that

$$\delta_h(x) = \frac{1}{h} W(x/h). \tag{45}$$

Theorem 7.2.1 from [6] gives sufficient conditions for the discrete moment conditions in terms of the behavior of the Fourier transform of W, which we restate here without proof.

Theorem 6 (continuous moment conditions [6]). *Consider the approximation*

$$f_{\text{app}}(\bar{x}) = \sum_{j \in \mathbb{Z}} W(j - \bar{x}) f(j). \tag{46}$$

Suppose that W decays sufficiently quickly, i.e., $|W(x)| \le A \exp(-B|x|)$ for some constants A and B. Then, the interpolation formula is of degree q if the following two conditions hold:

(1) *The function $\widetilde{W}(k) - 1$ has a zero of order q at $k = 0$.*

(2) *The function $\widetilde{W}(k)$ has a zero of order q at $k = 2\pi j$ for integer $j \neq 0$.*

We note that the first condition in Theorem 6 is equivalent to the continuous moment conditions in (29) and the second condition arises from the periodic summation of the spectrum of W due to sampling.

For computation purposes, it is desirable that the support of W be as small as possible. Due to a theorem of Tornberg and Engquist, we have the following minimum bound on this support.

Theorem 7 (minimum support for an approximate delta function [19]). *There exists a function $W \in Q^q$ if and only if the support of W contains the interval $[-q/2, q/2]$. Furthermore, for each choice of q, there is a unique W that achieves this minimum support though it is not, in general, smooth (or even continuous).*

In other words, Theorem 7 says that the support of W centered at 0 must cover at least $q+1$ points in the discrete grid such that the support of W with arbitrary center covers at least q points in the discrete grid. This ensures an adequate number of degrees of freedom to satisfy the discrete moment conditions.

Define the B-splines via the recursion

$$M_q = M_{q-1} * M_1, \quad M_1 = \chi_{[-1/2,1/2]}, \tag{47}$$

where $\chi_{[a,b]}$ is the indicator function of the interval $[a, b]$. It is evident that $M_q \in C^{q-2}$ is supported on the interval $[-q/2, q/2]$, and it is well known that its Fourier transform is given by

$$\widetilde{M}_q(k) = \left(\frac{\sin(k/2)}{k/2}\right)^q, \tag{48}$$

which has zeros of order q at nonzero integer multiples of 2π. Unfortunately, $\widetilde{M}_q(k) - 1$ has zeros of only order 2 at $k = 0$, restricting B-splines to only second-order approximations of the discrete delta distribution [15].

Based on these facts, we suppose for simplicity that q is even[1] and introduce the ansatz

$$\widetilde{W}_q(k) = \sum_{p=0}^{q/2-1} a_{2p} k^{2p} \widetilde{M}_q(k). \tag{49}$$

Because $\widetilde{M}_q(k)$ decays as $1/k^q$, we see that $\widetilde{W}_q(k)$ decays at least as fast as $1/k^2$, ensuring that $W_q(x)$ is continuous. Furthermore, it is evident that $\widetilde{W}_q(k)$ still has zeros of order q at $j2\pi$ for integer $j \neq 0$ regardless of the choice of coefficients a_p. Finally, we see that $\widetilde{W}_q(k)$ is real and even, therefore leading to a $W_q(x)$ that is real and symmetric. It remains to choose these coefficients such that $\widetilde{W}_q(k) - 1$ has zeros of order q at $k = 0$.

Let the Taylor expansion of $\widetilde{M}_q(k)$ about 0 be given by

$$\widetilde{M}_q(k) = \sum_{p=0}^{q/2-1} b_{2p} k^{2p} + \mathcal{O}(k^q), \tag{50}$$

[1] A similar argument holds for odd q, but the resulting approximant is not continuous.

where we note that $\widetilde{M}_q(k)$ is an even function and thus all odd coefficients are necessarily zero. Then, the Taylor expansion of $\widetilde{W}_q(k)$ about 0 is given by

$$\widetilde{W}_q(k) = \sum_{m=0}^{q/2-1} \left(\sum_{p=0}^{m} a_{2p} b_{2m-2p} \right) k^{2m} + \mathcal{O}(k^q), \tag{51}$$

where we see that the coefficients are given by a convolutional formula. To ensure zeros of the appropriate order, we would like to choose a_p such $a_0 b_0 = 1$ and the rest of the coefficients are 0. This leads to $q/2$ equations in $q/2$ unknowns in a triangular system of the form

$$\begin{bmatrix} a_0 & 0 & 0 & \cdots & 0 \\ a_2 & a_0 & 0 & \cdots & 0 \\ a_4 & a_2 & a_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{q-2} & a_{q-4} & a_{q-6} & \cdots & a_0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_2 \\ b_4 \\ \vdots \\ b_{q-2} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{52}$$

It is easy to verify that, for the B-splines, $a_0 = 1$ and thus this system is nonsingular, yielding a unique set of coefficients that lead to a $\widetilde{W}_q(k)$ satisfying the conditions of Theorem 6.

Now that we see we can attain a Fourier representation of an appropriate kernel of the form in (49), it remains to transform back to the spatial domain. However, by simple properties of the Fourier transform, we have that

$$W_q(x) = \sum_{p=0}^{q/2-1} a_{2p} (-1)^p M_q^{(2p)}(x); \tag{53}$$

i.e., we are simply taking a linear combination of the B-spline and its derivatives, leading once again to a piecewise polynomial spline supported on $[-q/2, q/2]$.

For completeness, we give the approximants W_q for $q = 4, 6$ produced by this method:

$$W_4(x) = \begin{cases} \frac{1}{2}|x|^3 - |x|^2 - \frac{1}{2}|x| + 1, & |x| \in [0, 1], \\ -\frac{1}{6}|x|^3 + |x|^2 - \frac{11}{6}|x| + 1, & |x| \in [1, 2], \\ 0, & \text{else.} \end{cases} \tag{54}$$

$$W_6(x) = \begin{cases} -\frac{1}{12}|x|^5 + \frac{1}{4}|x|^4 + \frac{5}{12}|x|^3 - \frac{5}{4}|x|^2 - \frac{1}{3}|x| + 1, & |x| \in [0, 1], \\ \frac{1}{24}|x|^5 - \frac{3}{8}|x|^4 + \frac{25}{24}|x|^3 - \frac{5}{8}|x|^2 - \frac{13}{12}|x| + 1, & |x| \in [1, 2], \\ -\frac{1}{120}|x|^5 + \frac{1}{8}|x|^4 - \frac{17}{24}|x|^3 + \frac{15}{8}|x|^2 - \frac{137}{60}|x| + 1, & |x| \in [2, 3], \\ 0, & \text{else.} \end{cases} \tag{55}$$

We note that the first coincides with the "k-point central interpolation formula" for $k = 4$ described, e.g., by Schoenberg [18]. In fact, discrete delta distributions

matching moment conditions to order q correspond directly with interpolation kernels on uniform grids that exactly integrate polynomials of order less than q [19], so the fact that both methods achieve the minimum support size of $[-q/2, q/2]$ means they are one and the same by Theorem 7. Finally, we remark that different forms of the ansatz in (49) can lead to kernels with slightly larger support but higher degrees of smoothness, if that is desired.

Acknowledgments

This research is supported by the Office of Advanced Scientific Computing Research of the U.S. Department of Energy under Contract Number DE-AC02-05CH11231. In addition, Minden is supported by a U.S. Department of Energy Computational Science Graduate Fellowship under grant number DE-FG02-97ER25308. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a user facility supported by the Office of Science of the U.S. Department of Energy under Contract Number DE-AC02-05CH11231.

Minden would like to thank A. Benson, A. Damle, and L. Ying for useful comments on early drafts of this manuscript.

References

[1] M. Adams, P. Colella, D. T. Graves, J. N. Johnson, H. S. Johansen, N. D. Keen, T. J. Ligocki, D. F. Martin, P. W. McCorquodale, D. Modiano, P. O. Schwartz, T. D. Sternberg, and B. Van Straalen, *Chombo software package for AMR applications: design document*, Tech. Report LBNL-6616E, Lawrence Berkeley National Laboratory, 2014.

[2] B. Alpert, L. Greengard, and T. Hagstrom, *An integral evolution formula for the wave equation*, J. Comput. Phys. **162** (2000), no. 2, 536–543. MR 1774266 Zbl 0966.65062

[3] K. Atkinson, *Numerical integration on the sphere*, J. Austral. Math. Soc. (B) **23** (1982), no. 3, 332–347. MR 642631 Zbl 0497.65010

[4] R. P. Beyer and R. J. LeVeque, *Analysis of a one-dimensional model for the immersed boundary method*, SIAM J. Numer. Anal. **29** (1992), no. 2, 332–364. MR 1154270 Zbl 0762.65052

[5] S. Chilton, *A fourth-order adaptive mesh refinement solver for Maxwell's Equations*, Ph.D. thesis, University of California, Berkeley, 2013. MR 3232238

[6] G.-H. Cottet and P. D. Koumoutsakos, *Vortex methods: theory and practice*, Cambridge University, Cambridge, 2000. MR 1755095

[7] B. Fornberg, *Generation of finite difference formulas on arbitrarily spaced grids*, Math. Comp. **51** (1988), no. 184, 699–706. MR 935077 Zbl 0701.65014

[8] M. Frigo and S. G. Johnson, *FFTW: an adaptive software architecture for the FFT*, Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing (Seattle, 1988), vol. 3, IEEE, Piscataway, NJ, 1998, pp. 1381–1384.

[9] T. Hagstrom, *High-resolution difference methods with exact evolution for multidimensional waves*, Appl. Numer. Math. **93** (2015), 114–122. MR 3323449 Zbl 1326.65107

[10] M. Hochbruck and A. Ostermann, *Exponential integrators*, Acta Numer. **19** (2010), 209–286. MR 2652783 Zbl 1242.65109

[11] R. W. Hockney, *The potential calculation and some applications*, Methods in computational physics (B. Alder, S. Fernbach, and M. Rotenberg, eds.), vol. 9: Plasma physics, Academic, New York, 1970, pp. 135–211.

[12] P. D. Lax, *Asymptotic solutions of oscillatory initial value problems*, Duke Math. J. **24** (1957), 627–646. MR 0097628 Zbl 0083.31801

[13] J.-R. Li and L. Greengard, *High order marching schemes for the wave equation in complex geometry*, J. Comput. Phys. **198** (2004), no. 1, 295–309. MR 2071396 Zbl 1052.65075

[14] P. McCorquodale, P. Colella, G. T. Balls, and S. B. Baden, *A local corrections algorithm for solving Poisson's equation in three dimensions*, Commun. Appl. Math. Comput. Sci. **2** (2007), 57–81. MR 2327083 Zbl 1133.65106

[15] J. J. Monaghan, *Extrapolating B splines for interpolation*, J. Comput. Phys. **60** (1985), no. 2, 253–262. MR 805872 Zbl 0588.41005

[16] C. S. Peskin, *The immersed boundary method*, Acta Numer. **11** (2002), 479–517. MR 2009378 Zbl 1123.74309

[17] R. D. Richtmyer and K. W. Morton, *Difference methods for initial-value problems*, 2nd ed., Interscience Tracts in Pure and Applied Mathematics, no. 4, Interscience, New York, 1967. MR 0220455 Zbl 0155.47502

[18] I. J. Schoenberg, *Contributions to the problem of approximation of equidistant data by analytic functions, A: On the problem of smoothing or graduation*, I. J. Schoenberg selected papers (C. de Boor, ed.), vol. 2, Birkhäuser, Boston, 1988, pp. 3–57.

[19] A.-K. Tornberg and B. Engquist, *Numerical approximations of singular source terms in differential equations*, J. Comput. Phys. **200** (2004), no. 2, 462–488. MR 2095274 Zbl 1115.76392

[20] J.-L. Vay, I. Haber, and B. B. Godfrey, *A domain decomposition method for pseudo-spectral electromagnetic simulations of plasmas*, J. Comput. Phys. **243** (2013), 260–268. MR 3064167

[21] B. Wang, G. H. Miller, and P. Colella, *A particle-in-cell method with adaptive phase-space remapping for kinetic plasmas*, SIAM J. Sci. Comput. **33** (2011), no. 6, 3509–3537. MR 2873252 Zbl 1232.76046

[22] G. B. Whitham, *Linear and nonlinear waves*, Wiley, New York, 1974. MR 0483954 Zbl 0373.76001

Received May 4, 2015. Revised January 28, 2016.

BORIS LO: bt.lo@berkeley.edu
Applied Science and Technology, University of California, Berkeley, Berkeley, CA 94720, United States

VICTOR MINDEN: vminden@stanford.edu
Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305, United States

PHILLIP COLELLA: pcolella@lbl.gov
Lawrence Berkeley National Laboratory, Berkeley, CA 94720, United States

and

Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA 94720, United States

COMM. APP. MATH. AND COMP. SCI.
Vol. 11, No. 2, 2016
dx.doi.org/10.2140/camcos.2016.11.171

ANALYSIS OF ESTIMATORS FOR ADAPTIVE KINETIC MONTE CARLO

DAVID ARISTOFF, SAMUEL T. CHILL AND GIDEON SIMPSON

Adaptive Kinetic Monte Carlo combines the simplicity of Kinetic Monte Carlo (KMC) with a saddle point search algorithm based on Molecular Dynamics (MD) in order to simulate metastable systems. Key to making Adaptive KMC effective is a stopping criterion for the saddle point search. In this work, we examine a criterion of S. T. Chill and G. Henkelman (*J. Chem. Phys.* **140** (2014), no. 21, 214110), which is based on the fraction of total reaction rate found instead of the fraction of observed saddles. The criterion uses the Eyring–Kramers law to estimate the reaction rate at the MD search temperature. We also consider a related criterion that remains valid when the Eyring–Kramers law is not. We examine the mathematical properties of both estimators and prove their mean square errors are well behaved, vanishing as the simulation continues to run.

1. Introduction

An outstanding problem in theoretical materials science and chemistry is how to reach laboratory time scales of microseconds (10^{-6} s) and longer using models, based on Molecular Dynamics (MD), which resolve the atomistic time scale of femtoseconds (10^{-15} s). Much of this scale separation is due to the presence of *metastable* regions in the configuration space of the system. In such regions, often defined by local minima of an energy landscape, the system stays close to a particular configuration, such as a local minima, before crossing into some other metastable region associated with a different configuration. Consequently, during much of a direct MD simulation, the system is close to one metastable region or another. It exhibits dynamics akin to a continuous time random walk on the set of metastable states, with comparatively long waiting times.

Since much of the physical significance of these systems is characterized by the sequence of visited metastable states and the time spent in each, there have been a variety of efforts to systematically coarse grain the MD trajectory into a more

Aristoff was supported by the National Science Foundation via the award NSF-DMS-1522398. Simpson was supported by US Department of Energy Award DE-SC0012733. He also thanks P. Hitczenko for helpful discussions.
MSC2010: 65C05, 65C20, 82C80.
Keywords: kinetic Monte Carlo, molecular dynamics, stopping time.

computationally efficient continuous time random walk. A. F. Voter has proposed three methods, Parallel Replica Dynamics, Hyperdynamics, and Temperature Accelerated Dynamics, which can overcome metastability through intelligent usage of the primitive Langevin dynamics [14; 16]. In recent years, significant effort has been made to understand and quantify the approximations in these methods and extend their applicability [2; 3; 1; 5; 11; 12; 15].

Another approach to the problem is Kinetic Monte Carlo (KMC), and this will be the focus of this work. Let us assume our system is governed by a potential energy $V(x)$, $x \in \mathbb{R}^d$ at inverse temperature β. Furthermore, we assume that we have partitioned configuration space into an at most countable set of metastable states, Ω_i, associated with local minima m_i of V. The system can go from metastable state i to metastable state j if there is a saddle point, s_{ij}, of $V(x)$ joining Ω_i and Ω_j. For conciseness, we will assume there is a single saddle point joining two given adjacent metastable states, though, in general, there may be multiple pathways.

In traditional KMC, before a simulation is run, one must identify the metastable states, their connectivity (i.e., which ones are joined by saddle points), and the reaction rates of each such connection. Given all of this information, KMC is very cheap to simulate. A single random number is generated and used to select one of the possible reactions, the system migrates into the new metastable region, and the algorithm repeats.

Unfortunately, such complete details of the metastable states and their connectivity are, *a priori*, unavailable in all but the simplest low-dimensional systems. This has motivated the development of Adaptive Kinetic Monte Carlo (AKMC) [6; 17; 18]. In AKMC, the system starts in some metastable region Ω_i. Saddle points associated with Ω_i are then sought via a *saddle point search algorithm* that successively finds s_{ij}. Reaction rates for each such saddle can be estimated by the Eyring–Kramers law [8]:

$$k_{ij} = g_{ij} \exp[-\beta(V(s_{ij}) - V(m_i))], \tag{1-1}$$

where, writing λ_1 for the sole negative eigenvalue of $\nabla^2 V(s_{ij})$,

$$g_{ij} = \frac{|\lambda_1|}{\pi} \sqrt{\left| \frac{\det \nabla^2 V(m_i)}{\det \nabla^2 V(s_{ij})} \right|}.$$

Once a sufficient number of saddles associated with Ω_i have been identified, the problem is treated by using traditional KMC with the thus far identified reactions and their rates; this process then repeats in the next metastable region. Two things are needed to proceed with AKMC:

(1) a saddle point search algorithm;

(2) a stopping criterion.

In this work, we will consider the question of the stopping criterion, provided our saddle point search algorithm satisfies certain assumptions. Our analysis will focus on estimators similar to the one introduced by Chill and Henkelman [6]. We call these *Chill-type estimators*.

In [6], the authors searched for saddle points out of each metastable state using high-temperature MD. For concreteness, consider the Brownian dynamics in $\mathbb{R}^d$:

$$dX_t = -\nabla V(X_t)\,dt + \sqrt{2\beta^{-1}}\,dW_t. \tag{1-2}$$

The aim is to model the dynamics at low temperature $\beta = \beta^{\mathrm{lo}}$. Starting at $X_0 \in \Omega_i$, integrate (1-2) at a higher temperature $\beta = \beta^{\mathrm{hi}}$ (i.e., $\beta^{\mathrm{lo}} > \beta^{\mathrm{hi}}$) until the trajectory leaves Ω_i. Using the higher temperature β^{hi} allows an escape to occur more quickly. After the trajectory leaves Ω_i, one of the saddle points s_{ij} is identified with this pathway using, for instance, the nudged elastic band method [10; 9], and the low-temperature reaction rate is computed using (1-1) with $\beta = \beta^{\mathrm{lo}}$. This is then repeated, with a new initial condition chosen in Ω_i. Throughout, the cumulative simulation time is recorded.

Other saddle point search algorithms have been proposed, including the dimer method and the string method [13; 7]. In our analysis, the key property that we need to hold true for all of our search methods is the following. Let

$$N_{ij}(t) = \text{Number of times saddle } s_{ij} \text{ has been found by time } t. \tag{1-3}$$

Then for fixed i, during a saddle point search, the $N_{ij}(t)$ are independent, with respect to j, Poisson processes. We prove below that this holds for a carefully performed saddle point search via integration of (1-2).

This article is organized as follows. We describe the saddle point search in detail in Section 2, and prove some of its properties, including the above condition on $N_{ij}(t)$, in Section 3. In Section 4 we introduce stopping criteria for the saddle point search, and in Section 5 we analyze these criteria. Section 6 contains proofs of some of the estimates in Section 5. In Section 7 we make some concluding remarks.

2. Notation and saddle point search algorithm

Here and throughout (X_t) is Brownian dynamics, that is, a stochastic process satisfying (1-2). For simplicity we fix a single metastable set $\Omega \equiv \Omega_i$ and suppress the index i in all of our notation from the introduction. For our purposes, V is smooth, and Ω is an (open) basin of attraction of V with respect to the gradient dynamics $dy/dt = -\nabla V(y)$. We assume that $\partial\Omega$ is partitioned into finitely many disjoint (measurable) subsets, called *pathways* and labeled $1, 2, \ldots, N$, such that each pathway j contains a unique saddle point s_j of V. When (X_t) leaves Ω, it must exit through one of the pathways $1, 2, \ldots, N$.

The algorithm, as well as our analysis, depends heavily on the quasistationary distribution (QSD) for (X_t) in Ω, which we denote by ν. The QSD ν is a probability measure that is locally invariant for (X_t), in the sense that it is invariant conditionally on the event that (X_t) remains in Ω:

Definition 2.1. The QSD for (X_t) in Ω is a probability measure ν supported in Ω such that for all $t > 0$,

$$\nu(\cdot) = \mathbb{P}(X_t \in \cdot \mid X_0 \sim \nu,\ X_s \in \Omega \text{ for all } s \in [0, t]).$$

Of course ν depends on Ω, but for simplicity we do not indicate this explicitly. It has been shown [11] that ν exists, is unique, and satisfies

$$\nu(A) = \lim_{n\to\infty} \mathbb{P}(X_t \in A \mid X_s \in A \text{ for } s \in [0, t]), \qquad \text{for all } A \subset \Omega. \tag{2-1}$$

Moreover this convergence is exponentially fast, uniformly in A. Equation (2-1) leads to simple algorithms for sampling ν, based on the idea that a sample can be obtained from the endpoint of a trajectory of (X_t) that has remained in Ω for a sufficiently long time; see [5] for details.

We are now ready to state the high-temperature saddle point search algorithm. Versions of this algorithm have been used previously; see for instance [6] and references therein. The search runs at a user-specified "high" (inverse) temperature β^{hi}. Below we write ν for the QSD in Ω at temperature $\beta = \beta^{\text{hi}}$. We also write

$$H(t) = \begin{cases} 0, & t < 0, \\ 1, & t \geq 0 \end{cases}$$

for the Heaviside unit step function.

Algorithm 2.2. Set $N_j(t) \equiv 0$ for $t \geq 0$ and $j = 1, \ldots, N$. Let M be the current cycle of the algorithm, and t_{sim} the simulation clock. Initialize $M = 1$ and $t_{\text{sim}} = 0$, and iterate the following:

1. Generate a sample x_M from ν. During this step t_{sim} is stopped.
2. Starting at $X_0 = x_M$, evolve (X_t) at $\beta = \beta^{\text{hi}}$ until it first leaves Ω, say at time $t = \tau^{(M)}$ through pathway $I^{(M)}$. The simulation clock t_{sim} is running during this step, and the stopping criterion is continuously checked. If at some time t_{sim} the criterion is met, the algorithm stops.
3. If $I^{(M)} = j$, update $N_j(t) = N_j(t) + H(t - t_{\text{sim}})$ for $t \geq 0$ and record the saddle point s_j. Then update $M = M + 1$. During this step t_{sim} is stopped.

It is not necessary to know N, and the pathways can be given labels according to the order in which they are found. The simulation clock is cumulative, and it only increases in Step 2. In particular, during the M-th cycle of the algorithm, t_{sim} increases by $\tau^{(M)}$. The stopping criterion will be described in Section 4. Below

we write t_{sim} for the final value of the simulation clock in the algorithm, that is, its value when the simulation is stopped. To refer to a generic simulation clock time we write t. Thus, $0 \le t \le t_{\text{sim}}$ and when the algorithm stops, $N_j(t)$ is the number of times an exit through pathway j has been observed by time t. Below we write $N_j(t)$ for its final value when the algorithm stops. We will also use the following notation:

$$\chi_j(t) = \mathbb{1}_{N_j(t)\ge 1}, \qquad N(t) = \sum_{j=1}^{N} N_j(t). \tag{2-2}$$

That is, $\chi_j(t) = 1$ if an exit through pathway j has been observed at least once by time t, and is 0 otherwise; $N(t)$ is the total number of exits observed by time t.

3. Properties of the saddle point search

Our first result follows immediately from properties of the QSD established in [11].

Theorem 3.1. *Suppose that in step* 1 *in the M-th cycle of Algorithm 2.2, x_M is a random variable with distribution ν. Then*:

(i) $\tau^{(M)}$ *is exponentially distributed with mean* κ^{-1}: $\mathbb{P}(\tau^{(j)} > t) = \exp(-\kappa t)$.

(ii) $\tau^{(M)}$ *and* $I^{(M)}$ *are independent.*

Theorem 3.1 then leads to the following.

Theorem 3.2. *Suppose that in step* 1 *of Algorithm 2.2, $x_1, x_2, \ldots$ are iid with common distribution ν. Then*:

(i) $\{N(t)\}_{0\le t\le t_{\text{sim}}}$ *is a Poisson process with parameter* κ.

(ii) $\{N_j(t)\}_{0\le t\le t_{\text{sim}}}^{j=1,\ldots,N}$ *are independent Poisson processes with parameters*

$$\kappa_j := \kappa p_j, \qquad p_j := \mathbb{P}(I^{(1)} = j). \tag{3-1}$$

Proof. Let $(\tilde{N}(s))_{s\ge 0}$ be a Poisson process with parameter κ, which we denote by $\tilde{N}(s)$ for brevity. Label each arrival time of $\tilde{N}(s)$ with a pathway j according to the distribution p_j, independently of the other arrival times, and let $\tilde{N}_j(s)$ be the process with arrivals labeled by j. Then for $r, s \ge 0$ and $m_1, \ldots, m_N \ge 0$,

$$\begin{aligned}
&\mathbb{P}\Bigg(\bigcap_{j=1}^{N}\{\tilde{N}_j(r+s) - \tilde{N}_j(r) = m_j\}\Bigg) \\
&\qquad = \mathbb{P}\Bigg(N(r+s) - N(r) = \sum_{j=1}^{N} m_j\Bigg)\binom{m_1+\cdots+m_N}{m_1,\ldots,m_N}\prod_{j=1}^{N} p_j^{m_j} \\
&\qquad = \prod_{j=1}^{N} \frac{e^{-\kappa p_j s}(\kappa p_j s)^{m_j}}{m_j!}.
\end{aligned} \tag{3-2}$$

By summing over all $m_i \geq 0$ for $i \neq j$ in the last expression above, we see that for fixed $r, s \geq 0$, the increment $\tilde{N}_j(r+s) - \tilde{N}_j(r)$ is Poisson distributed with mean $\kappa p_j s$. $\tilde{N}_j(s)$ also inherits independent increments from $\tilde{N}(s)$. This shows that $\tilde{N}_j(s)$ is a Poisson process with parameter $\kappa_j = \kappa p_j$. Moreover, (3-2) shows that $\tilde{N}_j(s)$, $j = 1, \ldots, N$, are independent.

Let us now relate $(\tilde{N}(s))_{s\geq 0}$ with $(N(s))_{0\leq s\leq t_{\text{sim}}}$. For fixed $s \in [0, t_{\text{sim}}]$, the time marginal $N(s)$ is the largest m such that $\tau^{(1)} + \cdots + \tau^{(m)} \leq s$. Together with part (i) of Theorem 3.1, this shows that on $[0, t_{\text{sim}}]$, $(N(s))_{0\leq s\leq t_{\text{sim}}}$ and $(\tilde{N}(s))_{s\geq 0}$ are Poisson processes with the same law. By part (ii) of Theorem 3.1, it follows that the multivariate processes $(N_j(s))_{0\leq s\leq t_{sim}}^{j=1,\ldots,N}$ and $(\tilde{N}_j(s))_{0\leq s\leq t_{\text{sim}}}^{j=1,\ldots,N}$ have the same law. This establishes the result. □

4. Chill-type estimators and stopping criteria

The purpose of the high-temperature saddle point search (Algorithm 2.2) is to locate "enough" of the low-temperature rate corresponding to the metastable set Ω. More precisely, at a low temperature corresponding to $\beta = \beta^{\text{lo}}$, the first exit time of X_t from Ω is approximately exponentially distributed with mean $(k_1 + \cdots + k_N)^{-1}$, where $k_j = k_j^{\text{lo}}$ is given by the Eyring–Kramers law (1-1) at $\beta = \beta^{\text{lo}}$ (recall the subscript i has been suppressed). See [4] and references therein for rigorous results in this direction. The k_j are then exponential rates associated with leaving Ω through pathway j at low temperature β^{lo}. The proportion of low-temperature rate found by time t in Algorithm 2.2 is

$$R(t) := \frac{\sum_{j=1}^N \chi_j(t) k_j}{\sum_{j=1}^N k_j}. \tag{4-1}$$

The expected value of $R(t)$ is

$$\mathbb{E}[R(t)] = \bar{R}(t) := \frac{\sum_{j=1}^N p_j(t) k_j}{\sum_{j=1}^N k_j}, \tag{4-2}$$

where

$$p_j(t) := \mathbb{E}[\chi_j(t)] = 1 - \exp(-\kappa_j t). \tag{4-3}$$

Here κ_j is defined as in Theorem 3.2 at temperature $\beta = \beta^{\text{hi}}$. The idea behind Chill-type estimators is that when $R(t)$ is sufficiently close to 1, the high-temperature saddle point search can stop. There are two obstacles to this idea.

The first is that, at any time during Algorithm 2.2, it is unlikely that all saddle points have been found. This problem is remedied by replacing k_j in (4-1) with $\chi_j(t)k_j$, which is computable once pathway j has been found during the simulation. The second obstacle is that an exact formula for $p_j(t) := \mathbb{E}[\chi_j(t)]$ will not be known

in practice. Chill-type estimators overcome the latter obstacle by using one of the following approximations:

$$\begin{aligned} &\tilde{p}_j(t) := 1 - \exp[-k_j^{\mathrm{hi}} t], \qquad k_j^{\mathrm{hi}} \text{ given by Eyring–Kramers (1-1) at } \beta = \beta^{\mathrm{hi}}, \\ &\hat{p}_j(t) := 1 - \exp[-\hat{N}_j(t)], \quad \hat{N}_j(t) := \begin{cases} N_j(t), & N_j(t) \geq 2, \\ 0, & \text{else.} \end{cases} \end{aligned} \tag{4-4}$$

We have used the superscript hi to emphasize that the rate in (4-4) is computed at temperature β^{hi} (whereas k_j is computed at low temperature β^{lo}). Also note that $\tilde{p}_j(t)$ is a physical estimate of $\mathbb{E}[\chi_j(t)]$ based on Eyring–Kramers, while $\hat{p}_j(t)$ is a (biased) Monte Carlo estimator. From (4-4) we obtain the following estimators for $R(t)$:

$$\tilde{R}(t) := \frac{\sum_{j=1}^N \tilde{p}_j(t)\chi_j(t)k_j}{\sum_{j=1}^N \chi_j(t)k_j}, \qquad \hat{R}(t) := \frac{\sum_{j=1}^N \hat{p}_j(t)\chi_j(t)k_j}{\sum_{j=1}^N \chi_j(t)k_j}. \tag{4-5}$$

$R(t)$, $\tilde{R}(t)$, and $\hat{R}(t)$ are all random, while $\bar{R}(t)$ is deterministic. Both $\tilde{R}(t)$ and $\hat{R}(t)$ are explicitly computable at time t during the saddle point search. See [6] for further discussion of $\tilde{R}(t)$. To our knowledge $\hat{R}(t)$ has not appeared before in the literature. We emphasize that $\hat{R}(t)$ may be used at any temperature β^{hi}, while $\tilde{R}(t)$ is limited by the fact that it gives reasonable estimates of $R(t)$ only at (relatively low) temperatures where the Eyring–Kramers law holds.

After choosing $\tilde{R}(t)$ or $\hat{R}(t)$ as the preferred estimator, the stopping criterion can now be defined as follows: for a user-specified parameter $\epsilon > 0$, stop Algorithm 2.2 in Step 3 if and only if

$$\tilde{R}(t) > 1 - \epsilon \quad \text{or} \quad \hat{R}(t) > 1 - \epsilon, \tag{4-6}$$

respectively. In Section 5 we give rigorous estimates of the bias and variance of the estimators $\tilde{R}(t)$ and $\hat{R}(t)$. These estimates show that, as t increases, when the algorithm stops, on average at least $(1-\epsilon)\%$ of the low-temperature rate has been found.

5. Analysis

The approximation $\tilde{p}_j(t)$ of $p_j(t)$ is usually considered valid when

$$\beta^{\mathrm{hi}} \ll V(s_j) - V(m),$$

with m the minimizer of V in Ω. To the authors' knowledge, rigorous results are scarce except when

$$s_j = \operatorname{argmin}_{s_1,\ldots,s_N} V(s_j) - V(m);$$

see [4] and references therein. However, the following is a consequence of results in [2]:

Theorem 5.1. *Suppose $\Omega = (a, b)$ is an interval and V is a Morse potential. Then for each $t > 0$,*

$$\frac{1 - \tilde{p}_j(t)}{1 - p_j(t)} = 1 + O(1/\beta^{\mathrm{hi}}) \quad \text{as } \beta^{\mathrm{hi}} \to \infty, \quad j = 1, 2. \tag{5-1}$$

Proof. An examination of the proof of Theorem 4.1 of [2] shows that for $j = 1, 2$,

$$k_j^{\mathrm{hi}}/\kappa_j = 1 + O(1/\beta^{\mathrm{hi}}) \quad \text{as } \beta^{\mathrm{hi}} \to \infty,$$

where k_j^{hi} is as in (4-4), and κ_j is as in Theorem 3.2 at temperature $\beta = \beta^{\mathrm{hi}}$. The result follows. □

We next examine the approximation $\hat{p}(t)$ of $p(t)$.

Theorem 5.2. *Conditionally on $N(t) \geq 1$, $\hat{N}_j(t)$ is an unbiased estimator for $\kappa_j t$:*

$$\mathbb{E}[\hat{N}_j(t) \mid N(t) \geq 1] = \kappa_j t. \tag{5-2}$$

Also conditionally on $N(t) \geq 1$, $\hat{p}_j(t)$ is a conservative estimate of $p_j(t)$:

$$\mathbb{E}[\hat{p}_j(t) \mid N_j(t) \geq 1] \leq p_j(t). \tag{5-3}$$

Proof. Recall that $N_j(t)$ is a Poisson process with parameter κ_j. Thus,

$$\begin{aligned}\mathbb{E}[\hat{N}_j(t) \mid N_j(t) \geq 1] &= (1 - e^{-\kappa_j t})^{-1} \sum_{n=2}^{\infty} n \frac{(\kappa_j t)^n e^{-\kappa_j t}}{n!} \\ &= \frac{\kappa_j t}{1 - e^{-\kappa_j t}} \sum_{n=1}^{\infty} \frac{(\kappa_j t)^n e^{-\kappa_j t}}{n!} = \kappa_j t.\end{aligned}$$

Since $x \mapsto 1 - e^{-x}$ is a concave function, the second statement of the theorem follows from Jensen's inequality. □

The reason that we consider conditional expectations in Theorem 5.2 is that Algorithm 2.2 cannot stop before $N(t) \geq 1$. Thus, we want estimates conditioned on that event. We call $\hat{p}_j(t)$ a conservative estimate for $p_j(t)$ because it is a lower bound on average, so that using $\hat{p}_j(t)$ in place of $p_j(t)$ leads to a larger average stopping time for Algorithm 2.2.

Before proceeding we define, for real-valued random variables X and Y,

$$\mathrm{Bias}(X, Y) := \mathbb{E}[X - Y], \quad \mathrm{MSE}(X, Y) := \mathrm{Bias}(X, Y)^2 + \mathrm{Var}(X). \tag{5-4}$$

Observe that the mean square error is not symmetric in its arguments.

Theorem 5.3. *Write* $q_j(t) = 1 - p_j(t) = \exp[-\kappa_j t]$ *and* $K = k_1 + \cdots + k_N$. *For the estimator* $\tilde{R}(t)$,

$$|\mathrm{Bias}(\tilde{R}(t), R(t))| \le N \max_j |\mathrm{Bias}(\tilde{p}_j(t), p_j(t))| + \frac{K}{\min_j k_j} \bar{R}(t) \max_j q_j(t),$$

$$\mathrm{Var}(\tilde{R}(t)) \le 4 \frac{K^2}{\min_j k_j^2} \bar{R}(t)^2 \max_j q_j(t),$$

$$\mathrm{MSE}(\tilde{R}(t), R(t)) \le 2N^2 \max_j \mathrm{MSE}(\tilde{p}_j(t), p_j(t)) + \frac{K^2}{\min_j k_j^2} \big(2 \max_j q_j(t) + 4\big) \bar{R}(t)^2 \max_j q_j(t).$$

For the estimator $\hat{R}(t)$,

$$|\mathrm{Bias}(\hat{R}(t), R(t))| \le N \max_j |\mathrm{Bias}(\hat{p}_j(t), p_j(t))| + \frac{K}{\min_j k_j} \bar{R}(t) \max_j q_j(t),$$

$$\mathrm{Var}(\hat{R}(t)) \le \frac{2K^2}{\min_j k_j^2} \bar{R}(t)^2 \max_j q_j(t) + \big(1 + 2N^2 \max_j q_j(t)\big) \max_j \mathrm{Var}(\hat{p}_j(t)),$$

$$\mathrm{MSE}(\hat{R}(t), R(t)) \le \big(1 + N^2 + 2N^2 \max_j q_j(t)\big) \max_j \mathrm{MSE}(\hat{p}_j(t), p_j(t)) + \frac{4K^2}{\min_j k_j^2} \bar{R}(t)^2 (1 + \max_j q_j(t)) \max_j q_j(t).$$

Here, all maxima and minima are taken over $j \in \{1, \ldots, N\}$.

Proof. We give proofs in Section 6 below. □

We note that some of the bounds in Theorem 5.3 have been loosened so that simpler expressions are obtained. This will become clear in the derivation of the bounds in Section 6 below. We highlight that the bias is bounded by the bias of the estimate of $p_j(t)$, together with another term representing an "inherent" bias associated with $\bar{R}(t)$. This second term may be approximated by noting that $|\bar{R}(t)| < 1$ for all t and, due to Theorem 5.1, we expect $q_j(t)$ can be estimated by the known function $\tilde{p}_j(t)$ or $\hat{p}_j(t)$.

6. Estimates

In this section we give a proof of Theorem 5.3. Recall that $q_j(t) := 1 - p_j(t)$ and $K := \sum_{j=1}^N k_j$ is the total reaction rate. For brevity, we will sometimes suppress the t dependence in our expressions. Also, all sums are over $1, \ldots, N$ unless otherwise indicated.

6A. ***Preliminary calculations.*** Observe that

$$\mathrm{Bias}(\tilde{R}(t), R(t)) = \mathrm{Bias}(\tilde{R}(t), \bar{R}(t)), \quad \mathrm{MSE}(\tilde{R}(t), R(t)) = \mathrm{MSE}(\tilde{R}(t), \bar{R}(t)),$$

and similarly for $\hat{R}(t)$; this fact will be used below without comment. There are a few expressions that will show up repeatedly in the analyses of both $\tilde{R}$ and $\hat{R}$. We analyze them here for simplicity. Let

$$\xi_i = k_i + \sum_{m \neq i} k_m \chi_m \tag{6-1}$$

We make the following calculations:

$$k_i \leq \xi_i \leq K, \tag{6-2a}$$

$$\mathbb{E}[\xi_i] = k_i + \sum_{m \neq i} p_m k_m = K - \sum_{m \neq i} q_m k_m. \tag{6-2b}$$

A lower bound on this can be obtained from Jensen's inequality,

$$\mathbb{E}[\xi_i^{-1}] \geq \mathbb{E}[\xi_i]^{-1} = \frac{1}{K - \sum_{m \neq i} q_m k_m} \geq \frac{1}{K} + \frac{1}{K^2} \sum_{m \neq i} k_m q_m, \tag{6-3}$$

while an upper bound can be obtained from the Edmundson–Madansky inequality,

$$\mathbb{E}[\xi_i^{-1}] \leq \frac{1}{k_i} \frac{K - \mathbb{E}[\xi_i]}{K - k_i} + \frac{1}{K} \frac{\mathbb{E}[\xi_i] - k_i}{K - k_i} = \frac{1}{K} + \frac{1}{k_i K} \sum_{m \neq i} k_m q_m. \tag{6-4}$$

In the same way,

$$\mathbb{E}[\xi_i^{-2}] \geq \mathbb{E}[\xi_i]^{-2} = \frac{1}{\left(K - \sum_{m \neq i} q_m k_m\right)^2} \geq \frac{1}{K^2} + \frac{2}{K^3} \sum_{m \neq i} q_m k_m, \tag{6-5}$$

and

$$\mathbb{E}[\xi_i^{-2}] \leq \frac{1}{k_i^2} \frac{K - \mathbb{E}[\xi_i]}{K - k_i} + \frac{1}{K^2} \frac{\mathbb{E}[\xi_i] - k_i}{K - k_i} = \frac{1}{K^2} + \frac{K + k_i}{k_i^2 K^2} \sum_{m \neq i} q_m k_m. \tag{6-6}$$

Therefore,

$$\mathrm{Var}(\xi_i^{-1}) \leq \left(\frac{K + k_i}{k_i^2 K^2} - \frac{2}{K^3} \right) \sum_{m \neq i} q_m k_m \leq \frac{2}{K k_i^2} \sum_{m \neq i} q_m(t) k_m, \tag{6-7}$$

where we have lost some of the estimate in the last inequality for the sake of conciseness.

6B. ***Estimates for $\tilde{R}$.*** Below it is useful to notice that

$$\tilde{R}(t) = \sum_{i=1}^{N} \frac{\tilde{p}_i(t) \chi_i(t) k_i}{k_i + \sum_{m \neq i} \chi_m(t) k_m} = \sum_i \frac{\tilde{p}_i \chi_i k_i}{\xi_i}. \tag{6-8}$$

6B1. *Bias.* We begin with the direct calculation

$$\begin{aligned}\mathbb{E}[\tilde{R}-\bar{R}] &= \sum_{i=1}^{N}\mathbb{E}\left[\frac{\chi_i\tilde{p}_i k_i}{\xi_i}-\frac{\chi_i k_i}{K}\right]\\ &= \sum_{i=1}^{N}(\tilde{p}_i-p_i)\mathbb{E}\left[\frac{\chi_i k_i}{\xi_i}\right]+\sum_{i=1}^{N}\mathbb{E}\left[\frac{\chi_i p_i k_i}{\xi_i}-\frac{\chi_i k_i}{K}\right]\\ &= \sum_{i=1}^{N}(\tilde{p}_i-p_i)\mathbb{E}\left[\frac{\chi_i k_i}{\xi_i}\right]+\sum_{i=1}^{N}\underbrace{\mathbb{E}\left[\frac{Kp_i}{\xi_i}-1\right]}_{\equiv b_i}\frac{p_i k_i}{K}.\end{aligned}$$

Using (6-3) and (6-4),

$$\frac{1}{K}\sum_{m\neq i}k_m q_m - q_i \le b_i \le \frac{1}{k_i}\sum_{m\neq i}k_m q_m - q_i.$$

Thus,

$$\left|\sum_{i=1}^{N} b_i\frac{p_i k_i}{K}\right| \le \sum_{i=1}^{N}\left(\sum_{j=1}^{N}\frac{k_j}{k_i}q_j\right)\frac{p_i(t)k_i}{K} \le \frac{K\max_j q_j(t)}{\min_j k_j}\bar{R}(t).$$

Combining the above expressions gives

$$|\mathrm{Bias}(\tilde{R}(t),\bar{R}(t))| \le N\max_i|\tilde{p}_i(t)-p_i(t)| + \frac{K\max_i q_i(t)}{\min_i k_i}\bar{R}(t). \tag{6-9}$$

6B2. *Variance.* For the variance, we first write

$$\tilde{R}-\mathbb{E}[\tilde{R}] = \sum_{i=1}^{N}\left(\frac{\chi_i}{\xi_i}-\mathbb{E}\left[\frac{\chi_i}{\xi_i}\right]\right)\tilde{p}_i k_i. \tag{6-10}$$

Hence,

$$\mathrm{Var}(\tilde{R}(t)) = \sum_{i,j=1}^{N} k_i k_j \tilde{p}_i\tilde{p}_j \underbrace{\mathrm{Cov}\left(\frac{\chi_i}{\xi_i},\frac{\chi_j}{\xi_j}\right)}_{\equiv v_{ij}}. \tag{6-11}$$

Since $v_{ij} \le \sqrt{v_{ii}}\sqrt{v_{jj}}$, it will be sufficient for us to analyze the diagonal terms. By Theorem 3.2, χ_i and ξ_i are independent. Thus

$$v_{ii} = \mathbb{E}[\chi_i]^2\,\mathrm{Var}(\xi_i^{-1}) + \mathbb{E}[\xi_i^{-1}]^2\,\mathrm{Var}(\chi_i) + \mathrm{Var}(\xi_i^{-1})\,\mathrm{Var}(\chi_i). \tag{6-12}$$

Using (6-6) and (6-7),

$$\begin{aligned} v_{ii} &\le p_i \operatorname{Var}(\xi_i^{-1}) + p_i q_i \mathbb{E}[\xi_i^{-2}] \\ &\le p_i \Big(\frac{K+k_i}{k_i^2 K^2} - \frac{2}{K^3} \Big) \sum_{m \neq i} q_m k_m + p_i q_i \Big(\frac{1}{K^2} + \frac{K+k_i}{k_i^2 K^2} \sum_{m \neq i} q_m k_m \Big) \\ &\le \frac{p_i q_i}{K^2} + \frac{4 p_i}{k_i^2 K} \sum_{m \neq i} q_m k_m \le \frac{4 p_i}{k_i^2} \max_j q_j \le \frac{4}{k_i^2} \max_j q_j(t) \\ &\le \frac{4}{\min_j k_j^2} \max_j q_j(t). \end{aligned} \tag{6-13}$$

We have made some sacrifices in the last inequalities in order to obtain a more concise expression. Consequently,

$$\operatorname{Var}(\tilde{R}(t)) \le \sum_{i,j=1}^{N} k_i k_j \tilde{p}_i(t) \tilde{p}_j(t) \sqrt{v_{ii}} \sqrt{v_{jj}} \le \frac{4K^2}{\min_i k_i^2} \bar{R}(t)^2 \max_i q_i(t). \tag{6-14}$$

6B3. *MSE.* Combining (6-9) and (6-14), we then obtain

$$\begin{aligned} \operatorname{MSE}(\tilde{R}(t), \bar{R}(t)) \le 2N^2 \max_i |\tilde{p}_i(t) - p_i(t)|^2 \\ + \frac{K^2}{\min_i k_i^2} \big(2 \max_i q_i(t) + 4 \big) \bar{R}(t)^2 \max_i q_i(t). \end{aligned} \tag{6-15}$$

In this calculation, we see that the mean square error may ultimately be dominated by how well the $\tilde{p}_i$ approximate the p_i.

6C. ***Estimates for $\hat{\boldsymbol{R}}$.*** We begin by noting that, since $\hat{p}_j(t) = 0$ if $\chi_j(t) \neq 1$,

$$\hat{R}(t) = \sum_j \frac{\hat{p}_j(t) k_j}{k_j + \sum_{m \neq j} \chi_m(t) k_j}. \tag{6-16}$$

6C1. *Bias.* We begin by writing

$$\hat{R} - \bar{R} = \sum_{i=1}^{N} (\hat{p}_i - p_i) \frac{k_i}{\xi_i} + \sum_{i=1}^{N} \frac{k_i p_i}{\xi_i} - \frac{k_i p_i}{K}, \tag{6-17}$$

so that, after taking an expectation,

$$\mathbb{E}[\hat{R} - \bar{R}] = \sum_{i=1}^{N} \mathbb{E}\left[(\hat{p}_i - p_i) \frac{k_i}{\xi_i} \right] + \sum_{i=1}^{N} \left(\mathbb{E}\left[\frac{K}{\xi_i} \right] - 1 \right) \frac{k_i p_i}{K}. \tag{6-18}$$

Hence,

$$|\operatorname{Bias}(\hat{R}(t), \bar{R}(t))| \le N \max_i |\operatorname{Bias}(\hat{p}_i(t), p(t))| + \frac{K}{\min_i k_i} \bar{R}(t) \max_i q_i(t), \tag{6-19}$$

and we see that the observed bias is controlled by the biases of the approximate probabilities, $\hat{p}_i$, and the inherent bias of the Chill-type estimators.

6C2. *Variance.* For the variance, we have

$$\operatorname{Var}(\hat{R}) = \sum_{i,j=1}^{N} k_i k_j \underbrace{\operatorname{Cov}\left(\frac{\hat{p}_i}{\xi_i}, \frac{\hat{p}_j}{\xi_j}\right)}_{\equiv \hat{v}_{ij}}. \tag{6-20}$$

As before, we only need to study the diagonal entries, and use Theorem 3.2 to obtain

$$\begin{aligned}
\hat{v}_{ii} &= \mathbb{E}[\hat{p}_i]^2 \operatorname{Var}(\xi_i^{-1}) + \mathbb{E}[\xi_i^{-1}]^2 \operatorname{Var}(\hat{p}_i) + \operatorname{Var}(\hat{p}_i) \operatorname{Var}(\xi_i^{-1}) \\
&\le \operatorname{Var}(\xi_i^{-1}) + \mathbb{E}[\xi_i^{-2}] \operatorname{Var}(\hat{p}_i) \\
&\le \frac{2}{\min_i k_i^2} \max_i q_i + \left(\frac{1}{K^2} + \frac{2}{\min_i k_i^2} \max_i q_i\right) \operatorname{Var}(\hat{p}_i) \\
&\le \frac{2}{\min_i k_i^2} \max_i q_i + \left(\frac{1}{K^2} + \frac{2}{\min_i k_i^2} \max_i q_i\right) \max_i \operatorname{Var}(\hat{p}_i).
\end{aligned} \tag{6-21}$$

We note that these estimates require full independence of $N_j(t)$ for $j = 1, \ldots, N$, not just independence of the $\chi_j(t)$. Now,

$$\operatorname{Var}(\hat{R}(t)) \le \frac{2K^2}{\min_i k_i^2} \bar{R}(t)^2 \max_i q_i(t) + (1 + 2N^2 \max_i q_i(t)) \max_i \operatorname{Var}(\hat{p}_i(t)). \tag{6-22}$$

6C3. *MSE.* We can therefore express the mean square error of estimator $\hat{R}$ as

$$\begin{aligned}
\operatorname{MSE}(\hat{R}(t), \bar{R}(t)) \le{}& \frac{4K^2}{\min_i k_i^2} \bar{R}(t)^2 (1 + \max_i q_i(t)) \max_i q_i(t) \\
&+ \left(1 + N^2 + 2N^2 \max_i q_i(t)\right) \max_i \operatorname{MSE}(\hat{p}_i(t), p_i(t)).
\end{aligned} \tag{6-23}$$

7. Discussion

We have considered three Chill-type estimators and shown them to be consistent. Their biases are small, relative to their variances, and thus we have good estimators of $R(t)$, the true fraction of the observed rate in the system. They represent a significant improvement over the original AKMC stopping criterion presented in [17]. Indeed, these prior approaches attempted to estimate the fraction of the saddles observed when, in fact, it is the fraction of the observed rate that is of fundamental importance.

As an example, we will compare the accuracy of both estimators using a test system that consists of saddle points s_j corresponding to potential energy barriers

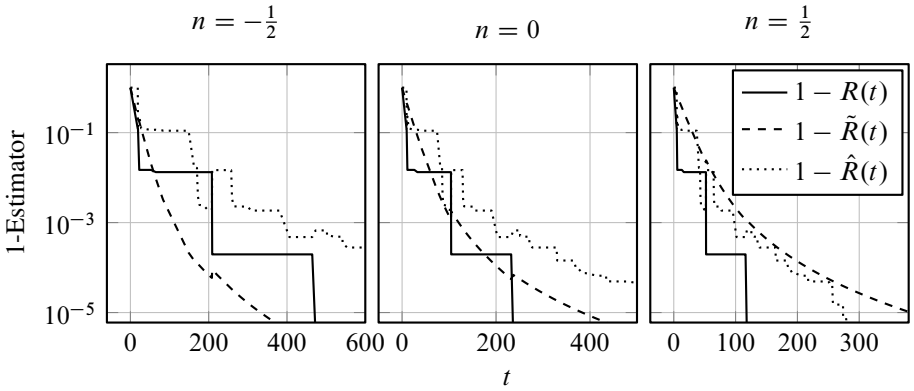


Figure 1. Comparison of the Chill-type estimators $\tilde{R}(t)$ and $\hat{R}(t)$ to the true expected proportion of the low-temperature rate found, $R(t)$, on a test system that can deviate from the Eyring–Kramer law.

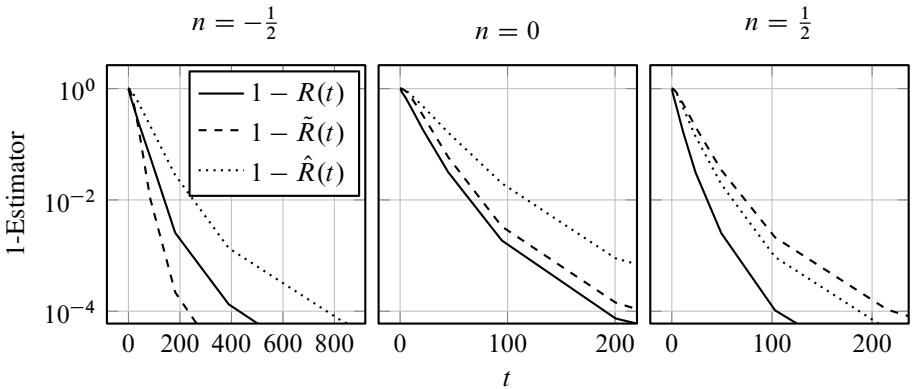


Figure 2. Comparison of the expected value of the Chill-type estimators $\tilde{R}(t)$ and $\hat{R}(t)$ to the true expected proportion of the low-temperature rate found, $R(t)$, on a test system that can deviate from the Eyring–Kramer law.

$V(s_j) - V(m) = 1 + \frac{4}{19}j$, for $j = 0, \ldots, 19$. The test system has rates that obey a modified Arrhenius equation with the form

$$\tilde{k}_j^{\mathrm{hi}} = \left(\frac{\beta^{\mathrm{lo}}}{\beta^{\mathrm{hi}}}\right)^n g_j \exp[\beta(V(s_j) - V(m))]. \tag{7-1}$$

Compare to Equation (1-1) (recall the subscript i has been suppressed). The variable n controls how the rates deviate from an unmodified Arrhenius rate law. When $n = 0$ the modified rates $\tilde{k}_j^{\mathrm{hi}}$ are equal to the unmodified rates k_j^{hi}, while when $\beta^{\mathrm{hi}} < \beta^{\mathrm{lo}}$, the modified rates are larger (resp. smaller) than the unmodified rates if $n > 0$ (resp. $n < 0$).

We use Algorithm 2.2 on the test system with modified rates $\tilde{k}_j^{\mathrm{hi}}$. This means $(N_j(t))_{0 \le t \le t_{\mathrm{sim}}}^{j=1,\ldots,N}$ are independent Poisson processes with parameters $\tilde{k}_j^{\mathrm{hi}}$. To compute

$R(t)$, we use (4-1) and sample $\chi_j(t)$ via (2-2). To compute $\tilde{R}(t)$ we use the unmodified Arrhenius rates k_j^{hi} in (4-4). For each of $R(t)$, $\tilde{R}(t)$ and $\hat{R}(t)$, the low-temperature rates $k_j = k_j^{\mathrm{lo}}$ used in (4-1) and (4-5) are the same. We take $g_j = 1$ for all j and $\beta^{\mathrm{hi}} = 2.5$, $\beta^{\mathrm{lo}} = 10.0$. The variable n was varied to compare the cases where the Eyring–Kramers rates k_j^{hi} underestimate $\left(n = \frac{1}{2}\right)$, overestimate $\left(n = -\frac{1}{2}\right)$, and provide an exact estimate ($n = 0$) of the modified rates $\tilde{k}_j^{\mathrm{hi}}$. Results are shown in Figures 1 and 2. The test system shows that $\tilde{R}(t)$ can overestimate $R(t)$ if the Eyring–Kramers rate deviates from the true rate at β^{hi}, while $\hat{R}(t)$ tends to provide a conservative estimate of $R(t)$.

References

[1] D. Aristoff, *The parallel replica method for computing equilibrium averages of Markov chains*, Monte Carlo Methods Appl. **21** (2015), no. 4, 255–273. MR 3433037 Zbl 1329.65005

[2] D. Aristoff and T. Lelièvre, *Mathematical analysis of temperature accelerated dynamics*, Multiscale Model. Simul. **12** (2014), no. 1, 290–317. MR 3176312 Zbl 1326.82018

[3] D. Aristoff, T. Lelièvre, and G. Simpson, *The parallel replica method for simulating long trajectories of Markov chains*, Appl. Math. Res. Express. (2014), no. 2, 332–352. MR 3266702 Zbl 1315.65010

[4] N. Berglund and S. Dutercq, *The Eyring–Kramers law for Markovian jump processes with symmetries*, J. Theor. Prob. (2015), 1–40.

[5] A. Binder, T. Lelièvre, and G. Simpson, *A generalized parallel replica dynamics*, J. Comput. Phys. **284** (2015), 595–616. MR 3303634

[6] S. T. Chill and G. Henkelman, *Molecular dynamics saddle search adaptive kinetic monte carlo*, J. Chem. Phys. **140** (2014), no. 21, 214110.

[7] W. E, W. Ren, and E. Vanden-Eijnden, *Simplified and improved string method for computing the minimum energy paths in barrier-crossing events*, J. Chem. Phys. **126** (2007), no. 16, 164103.

[8] P. Hanggi, P. Talkner, and M. Borkovec, *Reaction-rate theory: fifty years after Kramers*, Rev. Modern Phys. **62** (1990), no. 2, 251–341. MR 1056234

[9] G. Henkelman and H. Jónsson, *Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points*, J. Chem. Phys. **113** (2000), no. 22, 9978–9985.

[10] G. Henkelman, B. P. Uberuaga, and H. Jónsson, *A climbing image nudged elastic band method for finding saddle points and minimum energy paths*, J. Chem. Phys. **113** (2000), no. 22, 9901–9904.

[11] C. Le Bris, T. Lelièvre, M. Luskin, and D. Perez, *A mathematical formalization of the parallel replica dynamics*, Monte Carlo Methods Appl. **18** (2012), no. 2, 119–146. MR 2926765 Zbl 1243.82045

[12] T. Lelièvre and F. Nier, *Low temperature asymptotics for quasistationary distributions in a bounded domain*, Anal. PDE **8** (2015), no. 3, 561–628. MR 3353826 Zbl 1320.58021

[13] R. A. Olsen, G. J. Kroes, G. Henkelman, A. Arnaldsson, and H. Jónsson, *Comparison of methods for finding saddle points without knowledge of the final states*, J. Chem. Phys. **121** (2004), no. 20, 9776–9792.

[14] D. Perez, B. P. Uberuaga, Y. Shim, J. G. Amar, and A. F. Voter, *Accelerated molecular dynamics methods: introduction and recent developments*, Ann. Rep. Comp. Chem. **5** (2009), 79–98.

[15] G. Simpson and M. Luskin, *Numerical analysis of parallel replica dynamics*, ESAIM Math. Model. Numer. Anal. **47** (2013), no. 5, 1287–1314. MR 3100764 Zbl 1298.65016

[16] A. F. Voter, F. Montalenti, and T. C. Germann, *Extending the time scale in atomistic simulation of materials*, Ann. Rev. Materials Research **32** (2002), no. 1, 321–346.

[17] L. Xu and G. Henkelman, *Adaptive kinetic monte carlo for first-principles accelerated dynamics*, J. Chem. Phys. **129** (2008), no. 11, 114104.

[18] L. Xu, D. Mei, and G. Henkelman, *Adaptive kinetic monte carlo simulation of methanol decomposition on cu(100)*, J. Chem. Phys. **131** (2009), no. 24, 244520.

Received June 30, 2015.

DAVID ARISTOFF: aristoff@rams.colostate.edu
Department of Mathematics, Colorado State University, 221 Weber Hall, Fort Collins, CO 80523-1894, United States

SAMUEL T. CHILL: sam.chill@quantumwise.com
QuantumWise A/S, Austin, TX 78749, United States

GIDEON SIMPSON: simpson@math.drexel.edu
Department of Mathematics, Drexel University, Korman Center, 33rd and Market Streets, Philadelphia, PA 19104, United States

COMM. APP. MATH. AND COMP. SCI.
Vol. 11, No. 2, 2016
dx.doi.org/10.2140/camcos.2016.11.187

COMPARISON OF CONTINUOUS AND DISCRETE-TIME DATA-BASED MODELING FOR HYPOELLIPTIC SYSTEMS

FEI LU, KEVIN K. LIN AND ALEXANDRE J. CHORIN

We compare two approaches to the predictive modeling of dynamical systems from partial observations at discrete times. The first is continuous in time, where one uses data to infer a model in the form of stochastic differential equations, which are then discretized for numerical solution. The second is discrete in time, where one directly infers a discrete-time model in the form of a nonlinear autoregression moving average model. The comparison is performed in a special case where the observations are known to have been obtained from a hypoelliptic stochastic differential equation. We show that the discrete-time approach has better predictive skills, especially when the data are relatively sparse in time. We discuss open questions as well as the broader significance of the results.

1. Introduction

We examine the problem of inferring predictive stochastic models for a dynamical system, given partial observations of the system at a discrete sequence of times. This inference problem arises in applications ranging from molecular dynamics to climate modeling (see, e.g., [10; 12] and references therein). The observations may come from a stochastic or a deterministic chaotic system. This inference process, often called stochastic parametrization, is useful both for reducing computational cost by constructing effective lower-dimensional models, and for making prediction possible when fully resolved measurements of initial data and/or a full model are not available.

Typical approaches to stochastic parametrization start by identifying a continuous-time model, usually in the form of stochastic differential equations (SDEs), then discretizing the resulting model to make predictions. One difficulty with this standard approach is that it often leads to hypoelliptic systems [19; 22; 28], in which the noise acts on a proper subset of state space directions. As we will explain, this degeneracy can make parameter estimation for hypoelliptic systems particularly difficult [28; 33; 30], making the resulting model a poor predictor for the system at hand.

MSC2010: 62M09, 65C60.
Keywords: hypoellipticity, stochastic parametrization, Kramers oscillator, statistical inference, discrete partial data, NARMA.

Recent work [8; 21] has shown that fully discrete-time approaches to stochastic parametrization, in which one considers a discrete-time parametric model and infers its parameters from data, have certain advantages over continuous-time methods. In this paper, we compare the standard, continuous-time approach with a fully discrete-time approach, in a special case where the observations are known in advance to have been produced by a hypoelliptic system whose form is known, and only some parameters remain to be inferred. We hope that this comparison, in a relatively simple and well-understood context, will clarify some of the advantages and disadvantages of discrete-time modeling for dynamical systems. We note that our discussion here leaves in abeyance the question of what to do in cases where much less is known about the origin of the data; in general, there is no reason to believe that a given set of observations was generated by any stochastic differential equation or by a Markovian model of any kind.

A major difficulty in discrete modeling is the derivation of the structure, i.e., of the terms in the discrete-time model. We show that when the form of the differential equation giving rise to the data is known, one can deduce possible terms for the discrete model, but not necessarily the associated coefficients, from numerical schemes. Note that the use of this idea places the discrete and continuous models we compare on an equal footing, in that both approaches produce models directly derived from the assumed form of the model.

Model and goals. The specific hypoelliptic stochastic differential equations we work with have the form

$$\begin{aligned} dx_t &= y_t\,dt, \\ dy_t &= (-\gamma y_t - V'(x_t))\,dt + \sigma\,dB_t, \end{aligned} \tag{1-1}$$

where B_t is a standard Wiener process. When the potential V is quadratic, i.e.,

$$V(x) = \frac{\alpha}{2}x^2, \quad \alpha > 0,$$

we get a linear Langevin equation. When the potential has the form

$$V(x) = \frac{\beta}{4}x^4 - \frac{\alpha}{2}x^2, \quad \alpha, \beta > 0,$$

this is the Kramers oscillator [20; 31; 3; 15]. It describes the motion of a particle in a double-well potential driven by white noise, with x_t and y_t being the position and the velocity of the particle; $\gamma > 0$ is a damping constant. The white noise represents the thermal fluctuations of a surrounding "heat bath", the temperature of which is connected to γ and σ via the Einstein relation $T = \sigma^2/(2\gamma)$. This system is ergodic, with stationary density $p(x, y) \propto \exp(-(2\gamma/\sigma^2)(\frac{1}{2}y^2 + V(x)))$. It has multiple time scales and can be highly nonlinear, but is simple enough to

permit detailed numerical study. Parameter estimation for this system is also rather well-studied [28; 30]. These properties make (1-1) a natural example for this paper.

One of our goals is to construct a model that can make short-time forecasts of the evolution of the variable x based on past observations $\{x_{nh}\}_{n=1}^{N}$, where $h > 0$ is the observation spacing, in the situation where the parameters γ, α, β, and σ are unknown. (The variable y is not observed; hence, even when the parameters are known, the initial value of y is missing when one tries to solve the SDEs to make predictions.) We also require that the constructed model be able to reproduce long-term statistics of the data, e.g., marginals of the stationary distribution. In part, this is because the form of the model (either continuous or discrete-time) is generally unknown, and reproduction of long-term statistics provides a useful criterion for selecting a particular model. But even more important, in order for a model to be useful for tasks like data assimilation and uncertainty quantification, it must faithfully capture relevant statistics on time scales ranging from the short term (on which trajectorywise forecasting is possible) to longer time scales.

Our main finding is that the discrete-time approach makes predictions as reliably as the true system that gave rise to the data (which is of course unknown in general), even for relatively large observation spacings, while a continuous-time approach is only accurate when the observation spacing h is small, even in very low-dimensional examples such as ours.

Paper organization. We briefly review some basic facts about hypoelliptic systems in Section 2, including the parameter estimation technique we use to implement the continuous-time approach. In Section 3, we discuss the discrete-time approach. Section 4 presents numerical results, and in Section 5 we summarize our findings and discuss broader implications of our results. For the convenience of the reader, we collect a number of standard results about SDEs and their numerical solutions in the appendices.

2. Brief review of the continuous-time approach

2A. ***Inference for partially observed hypoelliptic systems.*** Consider a stochastic differential equation of the form

$$\begin{aligned} dX &= f(X, Y)\, dt, \\ dY &= a(X, Y)\, dt + b(X, Y)\, dW_t. \end{aligned} \tag{2-1}$$

Observe that only the Y equation is stochastically forced. Because of this, the second-order operator in the Fokker–Planck equation

$$\frac{\partial}{\partial t} p(x, y, t) = -\frac{\partial}{\partial x}[f(x, y)p(x, y, t)] - \frac{\partial}{\partial y}[a(x, y)p(x, y, t)] + \frac{1}{2}\frac{\partial^2}{\partial y^2}[b^2(x, y)p(x, y, t)] \tag{2-2}$$

for the time evolution of probability densities is not elliptic. This means that without any further assumptions on (2-1), the solutions of the Fokker–Planck equation, and hence the transition probability associated with the SDE, might be singular in the X direction. Hypoellipticity is a condition that guarantees the existence of smooth solutions for (2-2) despite this degeneracy. Roughly speaking, a system is hypoelliptic if the drift terms (i.e., the vector fields $f(x, y)$ and $a(x, y)$) help to spread the noise to all phase space directions, so that the system has a nondegenerate transition density. Technically, hypoellipticity requires certain conditions involving the Lie brackets of drift and diffusion fields, known as Hörmander's conditions [26]; when these conditions are satisfied, the system can be shown to possess smooth transition densities.

Our interest is in systems for which only discrete observations of x are available, and we use these observations to estimate the parameters in the functions f, a, and b. While parameter estimation for completely observed nondegenerate systems has been widely investigated (see e.g., [29; 33]), and there has been recent progress toward parameter estimation for partially observed nondegenerate systems [16], parameter estimation from discrete partial observations for hypoelliptic systems remains challenging.

There are three main categories of methods for parameter estimation (see, e.g., the surveys [32; 33]):

- Likelihood-type methods, where the likelihood is analytically or numerically approximated, or a likelihood-type function is constructed based on approximate equations. These methods lead to maximum likelihood estimators (MLE).
- Bayesian methods, in which one combines a prior with a likelihood, and one uses the posterior mean as estimator. Bayesian methods are important when the likelihood has multiple maxima. However, suitable priors may not always be available.
- Estimating function methods, or generalized moments methods, where estimators are found by estimating functions of parameters and observations. These methods generalize likelihood-type methods, and are useful when transition densities (and hence likelihoods) are difficult to compute. Estimating functions can be constructed using associated martingales or moments.

Because projections of Markov processes are typically not Markov, and the system is hypoelliptic, all three of the above approaches face difficulties for systems like (1-1): the likelihood function is difficult to compute either analytically or numerically, because only partial observations are available, and likelihood-type functions based on approximate equations often lead to biased estimators [11; 28; 30]. There are also no easily calculated martingales on which to base estimating functions [9].

There are two special cases that have been well-studied. When the system is linear, the observed process is a continuous-time autoregression process. Parameter estimation for this case is well-understood; see, e.g., the review papers [4; 7]. When the observations constitute an integrated diffusion (that is, $f(x, y) = y$ and the Y equation is autonomous, so that X is an integral of the diffusion process Y), consistent, asymptotically normal estimators are constructed in [9] using prediction-based estimating functions, and in [11] using a likelihood-type method based on Euler approximation. However, these approaches rely on the system being linear or the unobserved process being autonomous, and are not adapted to general hypoelliptic systems.

To our knowledge, for general hypoelliptic systems with discrete partial observation, only Bayesian-type methods [28] and a likelihood-type method [30] have been proposed when $f(x, y)$ is such that (2-1) can be written in the form of (1-1) by a change of variables. In [28] Euler and Itô–Taylor approximations are combined in a deterministic scan Gibbs sampler alternating between parameters and missing data in the unobserved variables. The reason for combining Euler and Itô–Taylor approximation is that Euler approximation leads to underestimated MLE of diffusion but is effective for drift estimation, whereas Itô–Taylor expansion leads to unbiased MLE of diffusion but is inappropriate for drift estimation. In [30] explicit consistent maximum likelihood-type estimators are constructed. However, all these methods require the observation spacing h to be small and the number of observations N to be large. For example, the estimators in [30] are only guaranteed to converge if, as $N \to \infty$, $h \to 0$ in such a way that $Nh^2 \to 0$ and $Nh \to \infty$. In practice, the observation spacing $h > 0$ is fixed, and large biases have been observed when h is not sufficiently small [28; 30]. We show in this paper that the bias can be so large that the prediction from the estimated system may be unreliable.

2B. ***Continuous-time stochastic parametrization.*** The continuous-time approach starts by proposing a parametric hypoelliptic system and estimating parameters in the system from *discrete partial observations*. In the present paper, the form of the hypoelliptic system is assumed to be known. Based on the Euler scheme approximation of the second equation in the system, Samson and Thieullen [30] constructed the likelihood-type function, or "contrast",

$$L_N(\theta) = \sum_{n=1}^{N-3} \frac{3}{2} \frac{[\hat{y}_{(n+2)h} - \hat{y}_{(n+1)h} + h(\gamma \hat{y}_{nh} + V'(x_{nh}))]^2}{h\sigma^2} + (N-3)\log\sigma^2,$$

where $\theta = (\gamma, \beta, \alpha, \sigma^2)$ and

$$\hat{y}_n = \frac{x_{(n+1)h} - x_{nh}}{h}. \tag{2-3}$$

Note that a shift in time in the drift term, i.e., the time index of $\gamma\hat{y}_{nh}+V'(x_{nh})$ is nh instead of $(n+1)h$, is introduced to avoid a $\sqrt{h}$ correlation between $\hat{y}_{(n+2)h}-\hat{y}_{(n+1)h}$ and $\gamma\hat{y}_{(n+1)h}+V'(x_{(n+1)h})$. Note also that there is a weighting factor $\frac{3}{2}$ in the sum, because the maximum likelihood estimator based on Euler approximation underestimates the variance (see, e.g., [11; 28]).

The estimator is the minimizer of the contrast:

$$\hat{\theta}_N = \arg\min_{\theta} L_N(\theta). \tag{2-4}$$

The estimator $\hat{\theta}_N$ converges to the true parameter value $\theta=(\gamma,\beta,\alpha,\sigma^2)$ under the condition that $h\to 0$, $Nh\to\infty$, and $Nh^2\to 0$. However, if h is not small enough, the estimator $\hat{\theta}_N$ can have a large bias (see [30] and the later sections), and the bias can be so large that the estimated system may have dynamics very different from the true system, and its prediction becomes unreliable.

Remark 2.1. In the case $V'(x)=\alpha x$, the Langevin system (1-1) is linear. The process $\{x_t : t\geq 0\}$ is a continuous-time autoregressive process of order 2, and there are various ways to estimate the parameters (see the review [5]), e.g., the likelihood method using a state-space representation and a Kalman filter [17], or methods for fitting discrete-time autoregression moving average (ARMA) models [27]. However, none of these approaches can be extended to nonlinear Langevin systems. In this section we focus on methods that work for nonlinear systems.

Once the parameters have been estimated, one numerically solves the estimated system to make predictions. In this paper, to make predictions for time $t>Nh$ (where N is the number of observations), we use the initial condition $(x_{Nh},\hat{y}_N)$ in solving the estimated system, with $\hat{y}_N$ being an estimate of y_{Nh} based on observations x. Since the system is stochastic, we use an "ensemble forecasting" method to make predictions. We start a number of trajectories from the same initial condition, and evolve each member of this ensemble independently. The ensemble characterizes the possible motions of the particle conditional on past observations, and the ensemble mean provides a specific prediction. For the purpose of short-term prediction, the estimated system can be solved with small time steps; hence, a low order scheme such as the Euler scheme may be used.

However, in many practical applications, the true system is unknown [8; 21], and one has to validate the continuous-time model by its ability to reproduce the long-term statistics of data. For this purpose, one has to compute the ergodic limits of the estimated system. The Euler scheme may be numerically unstable when the system is not globally Lipschitz, and a better scheme such as implicit Euler (see, e.g., [23; 34; 24]) or the quasisymplectic integrator [25] is needed. In our study, the Euler scheme is numerically unstable, while the Itô–Taylor scheme of strong order 2.0 (Scheme C.2) produces long-term statistics close to those produced by the

implicit Euler scheme. We use the Itô–Taylor scheme, since it has the advantage of being explicit and was used in [28].

In summary, the continuous-time approach uses the following algorithm to generate a forecasting ensemble of trajectories.

Algorithm 2.2 (continuous-time approach). With data $\{x_{nh}\}_{n=1}^{N}$,

Step 1. estimate the parameters using (2-4),

Step 2. select a numerical scheme for the SDE, e.g., the Itô–Taylor scheme in the appendix, and

Step 3. solve the SDE (1-1) with estimated parameters, using small time steps dt and initial data $(x_{Nh}, (x_{Nh} - x_{Nh-h})/h)$, to generate the forecasting ensemble.

3. The discrete-time approach

3A. ***NARMA representation.*** In the discrete-time approach, the goal is to infer a discrete-time predictive model for x from the data. Following [8], we choose a discrete-time system in the form of a nonlinear autoregression moving average (NARMA) model of the form

$$X_n = \Phi_n + \xi_n, \tag{3-1}$$

$$\Phi_n := \mu + \sum_{j=1}^{p} a_j X_{n-j} + \sum_{k=1}^{r} b_k Q_k(X_{n-p:n-1}, \xi_{n-q:n-1}) + \sum_{j=1}^{q} c_j \xi_{n-j}, \tag{3-2}$$

where p is the order of the autoregression, q is the order of the moving average, and the Q_k are given nonlinear functions (see below) of $(X_{n-p:n-1}, \xi_{n-q:n-1})$. Here $\{\xi_n\}$ is a sequence of i.i.d. Gaussian random variables with mean 0 and variance c_0^2 (denoted by $\mathcal{N}(0, c_0^2)$). The numbers p, q, and r as well as the coefficients a_j, b_j, and c_j are to be determined from data.

A main challenge in designing NARMA models is the choice of the functions Q_k, a process we call "structure selection" or "structure derivation". Good structure design leads to models that fit data well and have good predictive capabilities. Using too many unnecessary terms, on the other hand, can lead to overfitting or inefficiency, while too few terms can lead to an ineffective model. As before, we assume that a parametric family containing the true model is known, and we show that suitable structures for NARMA can be derived from numerical schemes for solving SDEs. We propose the following practical criteria for structure selection: the model should be numerically stable, we select the model that makes the best predictions (in practice, the predictions can be tested using the given data), and the large-time statistics of the model should agree with those of the data. These criteria are not sufficient to uniquely specify a viable model, and we shall return to this issue when we discuss the numerical experiments.

Once the Q_k have been chosen, the coefficients (a_j, b_j, c_j) are estimated from data using the following conditional likelihood method. Conditional on $\xi_1, \ldots, \xi_m$, the log-likelihood of $\{X_n = x_{nh}\}_{n=m+1}^N$ is

$$L_N(\vartheta \mid \xi_1, \ldots, \xi_m) = \sum_{n=m+1}^{N} \frac{(X_n - \Phi_n)^2}{2c_0^2} + \frac{N-q}{2} \log c_0^2, \tag{3-3}$$

where $m = \max\{p, q\}$ and $\vartheta = (a_j, b_j, c_j, c_0^2)$, and Φ_n is defined in (3-2). The log-likelihood is computed as follows. Conditionally on given values of $\{\xi_1, \ldots, \xi_m\}$, one can compute Φ_{m+1} from data $\{X_n = x_{nh}\}_{n=1}^m$ using (3-2). With the value of ξ_{m+1} following from (3-1), one can then compute Φ_{m+2}. Repeating this recursive procedure, one obtains the values of $\{\Phi_n\}_{n=m+1}^N$ that are needed to evaluate the log-likelihood. The estimator of the parameter $\vartheta = (a_j, b_j, c_j, c_0^2)$ is the minimizer of the log-likelihood

$$\hat{\vartheta}_N = \arg\min_{\vartheta} L_N(\vartheta \mid \xi_1, \ldots, \xi_m).$$

If the system is ergodic, the conditional maximum likelihood estimator $\hat{\vartheta}_N$ can be proved to be consistent (see, e.g., [1; 13]), which means that it converges almost surely to the true parameter value as $N \to \infty$. Note that the estimator requires the values of $\xi_1, \ldots, \xi_m$, which are in general not available. But ergodicity implies that if N is large, $\hat{\vartheta}_N$ forgets about the values of $\xi_1, \ldots, \xi_m$ quickly anyway, and in practice, we can simply set $\xi_1 = \cdots = \xi_m = 0$. Also, in practice, we initialize the optimization with $c_1 = \cdots = c_q = 0$ and with the values of (a_j, b_j) computed by least squares.

Note that in the case $q = 0$, the estimator is the same as the nonlinear least-squares estimator. The noise sequence $\{\xi_n\}$ does not have to be Gaussian for the conditional likelihood method to work, so long as the expression in (3-3) is adjusted accordingly.

In summary, the discrete-time approach uses the following algorithm to a generate a forecasting ensemble.

Algorithm 3.1 (discrete-time approach). With data $\{x_{nh}\}_{n=1}^N$,

Step 1. find possible structures for NARMA,

Step 2. estimate the parameters in NARMA for each possible structure,

Step 3. select the structure that fits the data best, in the sense that it reproduces best the long-term statistics and makes the best predictions, and

Step 4. use the resulting model to generate a forecasting ensemble.

3B. ***Structure derivation for the linear Langevin equation.*** The main difficulty in the discrete-time approach is the derivation of the structure of the NARMA model. In this section we discuss how to derive this structure from the SDEs, first in the linear case.

For the linear Langevin equation, the discrete-time system should be linear. Hence, we set $r = 0$ in (3-1) and obtain an ARMA(p, q) model. The linear Langevin equation

$$\begin{aligned} dx &= y\,dt, \\ dy &= (-\gamma y - \alpha x)\,dt + \sigma\,dB_t \end{aligned} \tag{3-4}$$

can be solved analytically. The solution x_t at discrete times satisfies

$$x_{(n+2)h} = a_1 x_{(n+1)h} + a_2 x_{nh} - a_{22} W_{n+1,1} + W_{n+2,1} + a_{12} W_{n+1,2} \tag{3-5}$$

(see Appendix A), where $\{W_{n,i}\}$ are defined in (A-1), and

$$a_1 = \operatorname{trace}(e^{\boldsymbol{A}h}), \quad a_2 = -e^{-\gamma h}, \quad a_{ij} = (e^{\boldsymbol{A}h})_{ij} \quad \text{for } \boldsymbol{A} = \begin{pmatrix} 0 & 1 \\ -\alpha & -\gamma \end{pmatrix}. \tag{3-6}$$

The process $\{x_{nh}\}$ defined in (3-5) is, strictly speaking, not an ARMA process (see Appendix B for all relevant, standard definitions used in this section), because $\{W_{n,1}\}_{n=1}^{\infty}$ and $\{W_{n,2}\}_{n=1}^{\infty}$ are not linearly dependent and would require at least two independent noise sequences to represent, while an ARMA process requires only one. However, as the following proposition shows, there is an ARMA process with the same distribution as the process $\{x_{nh}\}$. Since the minimum mean-square-error state predictor of a stationary Gaussian process depends only on its autocovariance function (see, e.g., [6, Chapter 5]), an ARMA process equal in distribution to the discrete-time Langevin equation is what we need here.

Proposition 3.2. *The* ARMA(2, 1) *process*

$$X_{n+2} = a_1 X_{n+1} + a_2 X_n + W_n + \theta_1 W_{n-1}, \tag{3-7}$$

where a_1 and a_2 are given in (3-6) and the $\{W_n\}$ are i.i.d. $\mathcal{N}(0, \sigma_W^2)$, is the unique process in the family of invertible ARMA processes that has the same distribution as the process $\{x_{nh}\}$. Here σ_W^2 and θ_1 ($\theta_1 < 1$ so that the process is invertible) satisfy the equations

$$\begin{aligned} \sigma_W^2(1 + \theta_1^2 + \theta_1 a_1) &= \gamma_0 - \gamma_1 a_1 - \gamma_2 a_2, \\ \sigma_W^2 \theta_1 &= \gamma_1(1 - a_2) - \gamma_0 a_1, \end{aligned}$$

where $\{\gamma_j\}_{j=0}^2$ are the autocovariances of the process $\{x_{nh}\}$ and are given in Lemma A.1.

Proof. Since the stationary process $\{x_{nh}\}$ is a centered Gaussian process, we only need to find an ARMA(p, q) process with the same autocovariance function

as $\{x_{nh}\}$. The autocovariance function of $\{x_{nh}\}$, denoted by $\{\gamma_n\}_{n=0}^{\infty}$, is given by (see Lemma A.1)

$$\gamma_n = \gamma_0 \times \begin{cases} \dfrac{1}{\lambda_1 - \lambda_2}(\lambda_1 e^{\lambda_2 nh} - \lambda_2 e^{\lambda_1 nh}) & \text{if } \gamma^2 - 4\alpha \neq 0, \\ e^{\lambda_0 nh}(1 - \lambda_0 nh) & \text{if } \gamma^2 - 4\alpha = 0, \end{cases}$$

where (λ_1, λ_2) or λ_0 are the roots of the characteristic polynomial $\lambda^2 + \gamma\lambda + \alpha = 0$ of the matrix $\boldsymbol{A}$ in (3-6).

On the other hand, the autocovariance function of an ARMA(p, q) process

$$X_n - \phi_1 X_{n-1} - \cdots - \phi_p X_{n-p} = W_n + \theta_1 W_{n-1} + \cdots + \theta_q W_{n-q},$$

denoted by $\{\gamma(n)\}_{n=0}^{\infty}$, is given by (see (B-4))

$$\gamma(n) = \sum_{i=1}^{k} \sum_{j=0}^{r_i - 1} \beta_{ij} n^j \zeta_i^{-n} \quad \text{for } n \geq \max\{p, q+1\} - p,$$

where $\{\zeta_i : i = 1, \ldots, k\}$ are the distinct zeros of $\phi(z) := 1 - \phi_1 z - \cdots - \phi_p z^p$, and r_i is the multiplicity of ζ_i (hence $\sum_{i=1}^{k} r_i = p$), and $\{\beta_{ij}\}$ are constants.

Since $\{\gamma_n\}_{n=0}^{\infty}$ only provides two possible roots, $\zeta_i = e^{-\lambda_i h}$ or $\zeta_i = e^{-\lambda_0 h}$ for $i = 1, 2$, the order must be $p = 2$. From these two roots, one can compute the coefficients ϕ_1 and ϕ_2 in the ARMA$(2, q)$ process:

$$\phi_1 = \zeta_1^{-1} + \zeta_2^{-1} = \operatorname{trace}(e^{\boldsymbol{A}h}) = a_1, \qquad \phi_2 = -\zeta_1^{-1}\zeta_2^{-1} = -e^{-\gamma h} = a_2.$$

Since $\gamma_k - \phi_1\gamma_{k-1} - \phi_2\gamma_{k-2} = 0$ for any $k \geq 2$, we have $q \leq 1$. As $\gamma_1 - \phi_1\gamma_0 - \phi_2\gamma_1 \neq 0$, Example B.2 indicates that $q \neq 0$. Hence, $q = 1$ and the above ARMA$(2, 1)$ is the unique process in the family of invertible ARMA(p, q) processes that has the same distribution as $\{x_{nh}\}$. The equations for σ_W^2 and θ_1 follow from Example B.3. □

This proposition indicates that the discrete-time system for the linear Langevin system should be an ARMA$(2, 1)$ model.

Example 3.3. Suppose $\Delta := \gamma^2 - 4\alpha < 0$. Then the parameters in the ARMA$(2, 1)$ process (3-7) are given by $a_1 = 2e^{-(\gamma/2)h}\cos(\frac{1}{2}\sqrt{-\Delta}h)$ and $a_2 = -e^{-\gamma h}$ and

$$\theta_1 = \frac{c - a_1 - \sqrt{(c - a_1)^2 - 4}}{2}, \qquad \sigma_w^2 = \frac{\gamma_1(1 - a_2) - \gamma_0 a_1}{\theta_1},$$

where $c = \dfrac{\gamma_0 - \gamma_1 a_1 - \gamma_2 a_2}{\gamma_1(1 - a_2) - \gamma_0 a_1}$ and $\gamma_n = \dfrac{\sigma^2}{2\gamma\alpha}\left(\cos\dfrac{\sqrt{-\Delta}nh}{2} + \dfrac{\gamma}{\sqrt{-\Delta}}\sin\dfrac{\sqrt{-\Delta}nh}{2}\right)$ for $n \geq 0$.

Remark 3.4. The maximum likelihood estimators of ARMA parameters can also be computed using a state-space representation and a Kalman recursion (see, e.g., [6]).

This approach is essentially the same as the conditional likelihood method in our discrete-time approach.

Remark 3.5. The proposition indicates that the parameters in the linear Langevin equation can also be computed from the ARMA(2, 1) estimators, because from the proof we have $\gamma = -\ln(-a_2)/h = -\lambda_1 - \lambda_2$, $\alpha = \lambda_1\lambda_2$, and $\sigma^2 = 2\gamma\alpha\sigma_W^2$, where $\{\lambda_i : i = 1, 2\}$ satisfy that $\{e^{-\lambda_i h} : i = 1, 2\}$ are the two roots of $\phi(z) = 1 - a_1 z - a_2 z$.

3C. ***Structure derivation for the Kramers oscillator.*** For nonlinear Langevin systems, in general there is no analytical solution, so the approach of Section 3B cannot be used. Instead, we derive structures from the numerical schemes for solving stochastic differential equations. For simplicity, we choose to focus on explicit terms in a discrete-time system, so implicit schemes (in, e.g., [23; 34; 25]) are not suitable. Here we focus on deriving structures from two explicit schemes: the Euler–Maruyama scheme and the Itô–Taylor scheme of order 2.0; see Appendix C for a brief review of these schemes. As mentioned before, we expect our approach to extend to other explicit schemes, e.g., that of [2]. While we consider specifically (1-1), the method used in this section extends to situations when $f(x, y)$ is such that (2-1) can be rewritten in form (1-1) and its higher-dimensional analogs by a change of variables.

To warm up, we begin with the Euler–Maruyama scheme. Applying Scheme C.1 to the system (1-1), we find

$$\begin{aligned} x_{n+1} &= x_n + y_n h, \\ y_{n+1} &= y_n(1 - \gamma h) - hV'(x_n) + W_{n+1}, \end{aligned}$$

where $W_n = \sigma h^{1/2}\zeta_n$, with $\{\zeta_n\}$ an i.i.d. sequence of $\mathcal{N}(0, 1)$ random variables. Straightforward substitutions yield a closed system for x

$$x_n = (2 - \gamma h)x_{n-1} - (1 - \gamma h)x_{n-2} - h^2 V'(x_{n-2}) + hW_{n-1}.$$

Since $V'(x) = \beta x^3 - \alpha x$, this leads to the following possible structure for NARMA:

Model (M1). $$X_n = a_1 X_{n-1} + a_2 X_{n-2} + b_1 X_{n-2}^3 + \xi_n + \sum_{j=1}^{q} c_j \xi_{n-j} + \mu.$$

Next, we derive a structure from the Itô–Taylor scheme of order 2.0. Applying Scheme C.2 to the system (1-1), we find

$$\begin{aligned} x_{n+1} &= x_n + h(1 - 0.5\gamma h)y_n - 0.5h^2 V'(x_n) + Z_{n+1}, \\ y_{n+1} &= y_n[1 - \gamma h + 0.5\gamma^2 h^2 - 0.5h^2 V''(x_n)] \\ &\qquad - h(1 - 0.5\gamma h)V'(x_n) + W_{n+1} - \gamma Z_{n+1}, \end{aligned}$$

where $Z_n = \sigma h^{3/2}(\zeta_n + \eta_n/\sqrt{3})$, with $\{\eta_n\}$ being an i.i.d. $\mathcal{N}(0,1)$ sequence independent of $\{\zeta_n\}$. Straightforward substitutions yield a closed system for x:

$$\begin{aligned} x_n = {} & x_{n-1}[2-\gamma h+0.5\gamma^2h^2-h^2V''(x_{n-2})]-0.5h^2V'(x_{n-1})+Z_n \\ & +[1-\gamma h+0.5\gamma^2h^2-0.5h^2V''(x_{n-2})](-x_{n-2}+0.5h^2V'(x_{n-2})-Z_{n-1}) \\ & -h^2(1-0.5\gamma h)^2V'(x_{n-2})+h(1-0.5\gamma h)(W_{n-1}-\gamma Z_{n-1}). \end{aligned}$$

Note that W_n is of order $h^{1/2}$ and Z_n is of order $h^{3/2}$. Writing the terms in descending order, we obtain

$$\begin{aligned} x_n = {} & (2-\gamma h+0.5\gamma^2h^2)x_{n-1}-(1-\gamma h+0.5\gamma^2h^2)x_{n-2}+Z_n-Z_{n-1} \\ & +h(1-0.5\gamma h)W_{n-1}-0.5h^2V'(x_{n-1})+0.5h^2V''(x_{n-2})(x_{n-1}-x_{n-2}) \\ & +0.5\gamma h^3V'(x_{n-2})+0.5h^2V''(x_{n-2})Z_{n-1}-0.5h^4V''(x_{n-2})V'(x_{n-2}). \end{aligned} \tag{3-8}$$

This equation suggests that $p=2$ and $q=0$ or 1. The noise term $Z_n - Z_{n-1} + h(1-0.5\gamma h)W_{n-1}$ is of order $h^{1.5}$, and involves two independent noise sequences $\{\zeta_n\}$ and $\{\eta_n\}$; hence, the above equation for x_n is not a NARMA model. However, it suggests possible structures for NARMA models. In comparison to Model (M1), the above equation has different nonlinear terms of order h^2: $h^2V'(x_{n-1})$ and $h^2V''(x_{n-2})(x_{n-1}-x_{n-2})$; and has additional nonlinear terms of orders 3 and larger: $h^3V'(x_{n-2})$, $h^2Z_{n-1}V''(x_{n-2})$, and $h^4V''(x_{n-2})V'(x_{n-2})$. It is not clear which terms should be used, and one may be tempted to include as many terms as possible. However, this can lead to overfitting. Hence, we consider different structures by successively adding more and more terms, and select the one that fits data the best. Using the fact that $V'(x)=\beta x^3-\alpha x$, these terms lead to the following possible structures for NARMA (for the reader's convenience, we have underlined all higher-order terms derived from $V'(x)$).

Model (M2). $$X_n = a_1X_{n-1}+a_2X_{n-2}+b_1X_{n-1}^3 + \underline{b_2X_{n-2}^2(X_{n-1}-X_{n-2})}+\xi_n+\sum_{j=1}^{q}c_j\xi_{n-j}+\mu,$$

where b_1 and b_2 are of order h^2, and $q \geq 0$.

Model (M3). $$X_n = a_1X_{n-1}+a_2X_{n-2}+b_1X_{n-1}^3 + \underline{b_2X_{n-2}^2(X_{n-1}-X_{n-2})}+\underline{b_3X_{n-2}^3}+\xi_n+\sum_{j=1}^{q}c_j\xi_{n-j}+\mu,$$

where b_3 is of order h^3, and $q \geq 0$.

Model (M4). $$X_n = a_1X_{n-1}+a_2X_{n-2}+b_1X_{n-1}^3+\underline{b_2X_{n-2}^2X_{n-1}} + \underline{b_3X_{n-2}^3}+\underline{b_4X_{n-2}^5}+\underline{b_5X_{n-2}^2\xi_{n-1}}+\xi_n+\sum_{j=1}^{q}c_j\xi_{n-j}+\mu,$$

where b_4 is of order h^4, and b_5 is of order $h^{3.5}$, and $q \geq 1$.

From Models (M2)–(M4), the number of nonlinear terms increases as their order increases in the numerical scheme. Following [8; 21], we only use the form of the terms derived from numerical analysis, and not their coefficients; we estimate new coefficients from data.

4. Numerical study

We test the continuous and discrete-time approaches for data sets with different observation intervals h. The data are generated by solving the general Langevin equation (1-1) using a second-order Itô–Taylor scheme, with a small step size $dt = \frac{1}{1024}$, and making observations with time intervals $h = \frac{1}{32}$, $\frac{1}{16}$, and $\frac{1}{8}$; the value of time step dt in the integration has been chosen to be sufficiently small to guarantee reasonable accuracy. For each one of the data sets, we estimate the parameters in the SDE and in the NARMA models. We then compare the estimated SDE and the NARMA model by their ability to reproduce long-term statistics and to perform short-term prediction.

4A. ***The linear Langevin equation.*** We first discuss numerical results in the linear case. Both approaches start by computing the estimators. The estimator $\hat{\theta} = (\hat{\gamma}, \hat{\alpha}, \hat{\sigma})$ of the parameters (γ, α, σ) of the linear Langevin equation (3-4) is given by

$$\hat{\theta} = \underset{\theta=(\gamma,\alpha,\sigma)}{\arg\min} \left[\sum_{n=1}^{N-3} \frac{3}{2} \frac{[\hat{y}_{n+2} - \hat{y}_{n+1} + h(\gamma \hat{y}_n + \alpha x_n)]^2}{h\sigma^2} + (N-3)\log\sigma^2 \right], \quad (4\text{-}1)$$

where $\hat{y}_n$ is computed from data using (2-3).

Following (3-7), we use the ARMA(2, 1) model in the discrete-time approach:

$$X_{n+2} = a_1 X_{n+1} + a_2 X_n + W_n + \theta_1 W_{n-1}.$$

We estimate the parameters a_1, a_2, θ_1, and σ_W^2 from data using the conditional likelihood method of Section 3A.

First, we investigate the reliability of the estimators. One hundred simulated data sets are generated from (3-4) with true parameters $\gamma = 0.5$, $\alpha = 4$, and $\sigma = 1$, and with initial condition $x_0 = y_0 = \frac{1}{2}$ and time interval $[0, 10^4]$. The estimators, of (γ, α, σ) in the linear Langevin equation and of $(a_1, a_2, \theta_1, \sigma_W)$ in the ARMA(2, 1) model, are computed for each data set. Empirical mean and standard deviation of the estimators are reported in Table 1 for the continuous-time approach, and Table 2 for the discrete-time approach. In the continuous-time approach, the biases of the estimators grow as h increases. In particular, large biases occur for the estimators of γ: the bias of $\hat{\gamma}$ increases from 0.2313 when $h = \frac{1}{32}$ to 0.4879 when $h = \frac{1}{8}$, while the true value is $\gamma = 0.5$; similarly large biases were also noticed in [30]. In contrast,

Estimator	True value	$h = \frac{1}{32}$	$h = \frac{1}{16}$	$h = \frac{1}{8}$
$\hat{\gamma}$	0.5	0.7313 (0.0106)	0.9538 (0.0104)	1.3493 (0.0098)
$\hat{\alpha}$	4	3.8917 (0.0193)	3.7540 (0.0187)	3.3984 (0.0172)
$\hat{\sigma}$	1	0.9879 (0.0014)	0.9729 (0.0019)	0.9411 (0.0023)

Table 1. Mean and standard deviation of the estimators of the parameters (γ, α, σ) of the linear Langevin equation in the continuous-time approach, computed on 100 simulations.

Estimator	$h = \frac{1}{32}$	$h = \frac{1}{16}$	$h = \frac{1}{8}$
$\hat{a}_1$	1.9806	1.9539	1.8791
	1.9807 (0.0003)	1.9541 (0.0007)	1.8796 (0.0014)
$-\hat{a}_2$	0.9845	0.9692	0.9394
	0.9846 (0.0003)	0.9695 (0.0007)	0.9399 (0.0014)
$\hat{\theta}_1$	0.2681	0.2684	0.2698
	0.2667 (0.0017)	0.2680 (0.0025)	0.2700 (0.0037)
$\hat{\sigma}_W$	0.0043	0.0121	0.0336
	0.0043 (0.0000)	0.0121 (0.0000)	0.0336 (0.0001)

Table 2. Mean and (in parentheses) standard deviation of the estimators of the parameters $(a_1, a_2, \theta_1, \sigma_W)$ of the ARMA(2, 1) model in the discrete-time approach, computed on 100 simulations. The theoretical values (listed above the mean values) are computed from Proposition 3.2.

the biases are much smaller for the discrete-time approach. The "theoretical values" of a_1, a_2, θ_1, and σ_W^2 are computed analytically as in Example 3.3. Table 2 shows that the estimators in the discrete-time approach have negligible differences from the theoretical values.

In practice, the above test of the reliability of estimators cannot be performed, because one has only a single data set and the true system that generated the data is unknown.

We now compare the two approaches in a practical setting, by assuming that we are only given a single data set from discrete observations of a long trajectory on time interval $[0, T]$ with $T = 2^{17} \approx 1.31 \times 10^5$. We estimate the parameters in the SDE and the ARMA model, and again investigate the performance of the estimated SDE and ARMA model in reproducing long-term statistics and in predicting the short-term evolution of x. The long-term statistics are computed by time-averaging. The first half of the data set is used to compute the estimators, and the second half of the data set is used to test the prediction.

The long-term statistics, i.e., the empirical probability density function (PDF) and the autocorrelation function (ACF), are shown in Figure 1. For all three values

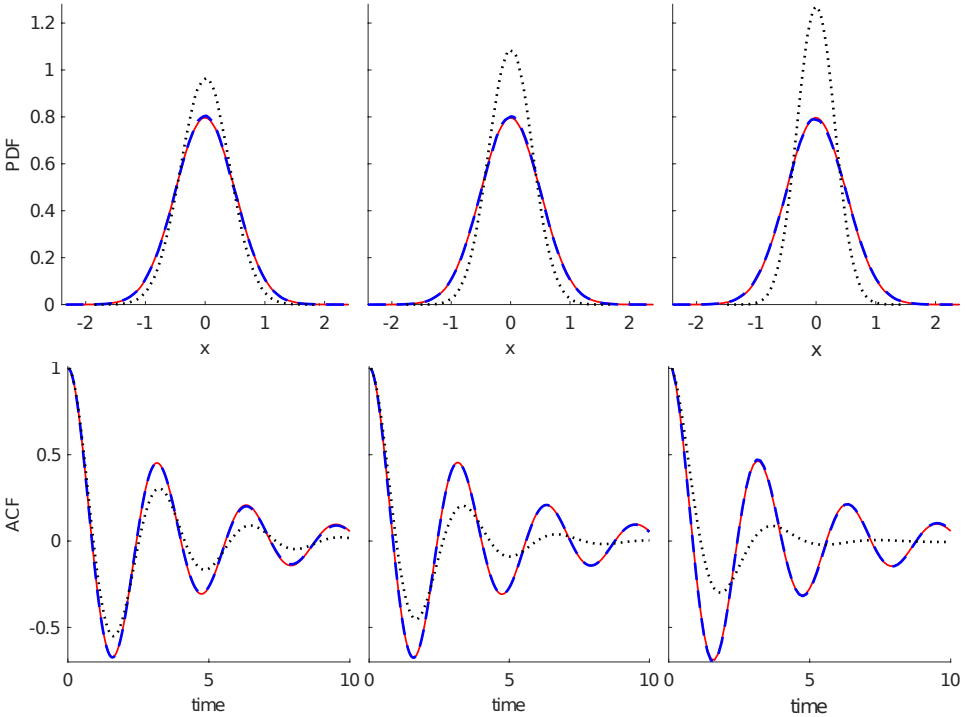


Figure 1. Empirical PDF and ACF of the ARMA(2, 1) models (blue dashed line) and the estimated linear Langevin system (black dotted line), in the cases $h = \frac{1}{32}$ (left), $h = \frac{1}{16}$ (center), and $h = \frac{1}{8}$ (right). The ARMA models reproduce the PDF and ACF almost perfectly (red solid line), much better than the estimated SDEs.

of h, the ARMA models reproduce the empirical PDF and ACF almost perfectly. The estimated SDEs miss the spread of the PDF and the amplitude of oscillation in the ACF, and these error become larger as h increases.

Next, we use an ensemble of trajectories to predict the motion of x. For each ensemble, we calculate the mean trajectory and compare it with the true trajectory from the data. We measure the performance of the prediction by computing the root-mean-square error (RMSE) of a large number of ensembles as follows: take N_0 short pieces of data from the second half of the long trajectory, denoted by $\{(x_{(n_i+1)h}, \ldots, x_{(n_i+K)h})\}_{i=1}^{N_0}$, where $n_i = Ki$. For each short piece of data $(x_{(n_i+1)h}, \ldots, x_{(n_i+K)h})$, we generate N_{ens} trajectories $\{(X_1^{i,j}, \ldots, X_K^{i,j})\}_{j=1}^{N_{\text{ens}}}$ using a prediction system (i.e., NARMA(p, q), the estimated Langevin system, or the true Langevin system), starting all ensemble members from the same several-step initial condition $(x_{(n_i+1)h}, \ldots, x_{(n_i+m)h})$, where $m = 2\max\{p, q\} + 1$. For NARMA(p, q) we start with $\xi_1 = \cdots = \xi_q = 0$. For the estimated Langevin system and the true Langevin system, we start with initial condition $(x_{(n_i+m)h}, \hat{y}_{n_i})$ with $\hat{y}_{n_i} = (x_{(n_i+m)h} - x_{(n_i+m-1)h})/h$ and solve the equations using the Itô–Taylor scheme

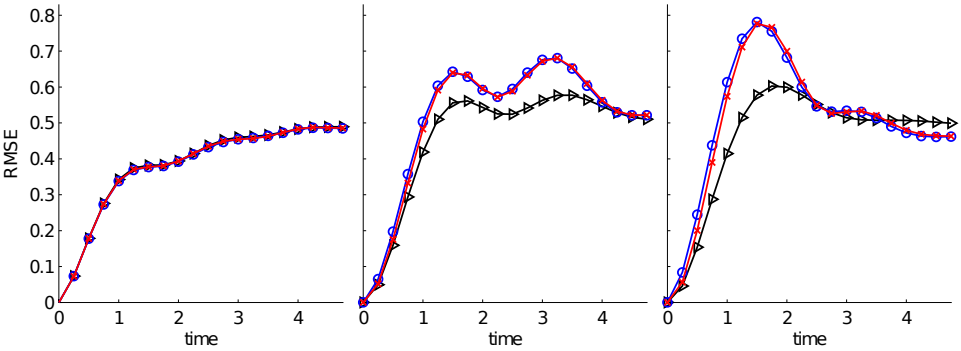


Figure 2. The linear Langevin system: RMSEs of 10^4 forecasting ensembles with size $N_{\text{ens}} = 20$, produced by the true system (black triangles), the system with estimated parameters (red x's), and the ARMA model (blue circles), in the cases $h = \frac{1}{32}$ (left), $h = \frac{1}{16}$ (center), and $h = \frac{1}{8}$ (right).

Estimator	True value	$h = \frac{1}{32}$	$h = \frac{1}{16}$	$h = \frac{1}{8}$
$\hat{\gamma}$	0.5	0.8726 (0.0063)	1.2049 (0.0057)	1.7003 (0.0088)
$\hat{\beta}$	0.3162	0.3501 (0.0007)	0.3662 (0.0007)	0.4225 (0.0009)
$\hat{\sigma}$	1	0.9964 (0.0014)	1.0132 (0.0027)	1.1150 (0.0065)

Table 3. Mean and standard deviation of the estimators of the parameters (γ, β, σ) of the Kramers equation in the continuous-time approach, computed on 100 simulations.

of order 2.0 with a time step $dt = \frac{1}{64}$ and record the trajectories every h/dt steps to get the prediction trajectories $(X_1^{i,j}, \ldots, X_K^{i,j})$.

We then calculate the mean trajectory for each ensemble $\overline{X}_k^i = (1/N_{\text{ens}}) \sum_{j=1}^{N_{\text{ens}}} X_k^{i,j}$, $k = 1, \ldots, K$. The RMSE measures, in an average sense, the difference between the mean ensemble trajectory and the true data trajectory:

$$\text{RMSE}(kh) := \left(\frac{1}{N_0} \sum_{i=1}^{N_0} |\overline{X}_k^i - x_{(n_i+k)h}|^2 \right)^{1/2}.$$

The RMSE measures the accuracy of the mean ensemble prediction; RMSE $= 0$ corresponds to a perfect prediction, and small RMSEs are desired.

The computed RMSEs for $N_0 = 10^4$ ensembles with $N_{\text{ens}} = 20$ are shown in Figure 2. The ARMA(2, 1) model reproduces almost exactly the RMSEs of the true system for all three observation step sizes, while the estimated system has RMSEs deviating from that of the true system as h increases. The estimated system has smaller RMSEs than the true system, because it underestimates the variance of the true process x_t (that is, $\hat{\sigma}^2/(2\hat{\alpha}\hat{\gamma}) < \sigma^2/(2\alpha\gamma)$) and because the means of x_t decay exponentially to 0. The steady increase in RMSE, even for the true system,

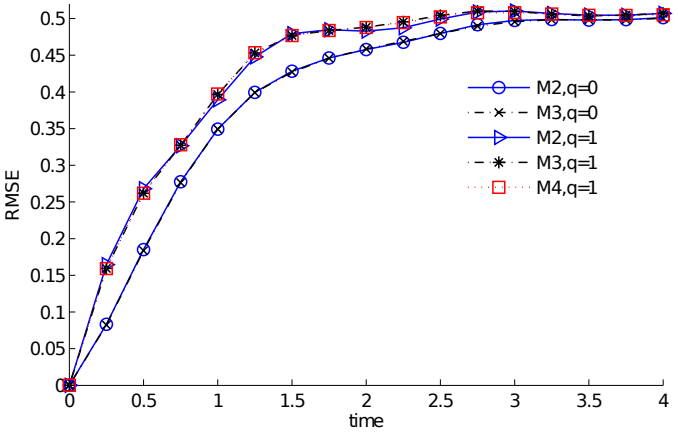


Figure 3. RMSEs of Models (M2), (M3), and (M4) with ensemble size $N_{\text{ens}} = 20$ in the case $h = \frac{1}{8}$. Models with $q = 1$ have larger RMSEs than the models with $q = 0$. In the case $q = 0$, Models (M2) and (M3) have almost the same RMSEs.

is entirely expected because the forecasting ensemble is driven by independent realizations of the forcing, as one cannot infer the white noise driving the system that originally generated the data.

4B. ***The Kramers oscillator.*** We consider the Kramers equation in the form

$$\begin{aligned} dx_t &= y_t\,dt, \\ dy_t &= (-\gamma y_t - \beta^{-2}x_t^3 + x_t)\,dt + \sigma\,dB_t, \end{aligned} \tag{4-2}$$

for which there are two potential wells located at $x = \pm\beta$.

In the continuous-time approach, the estimator $\hat{\theta} = (\hat{\gamma}, \hat{\beta}, \hat{\sigma})$ is given by

$$\hat{\theta} = \underset{\theta=(\gamma,\beta,\sigma)}{\arg\min}\left[\sum_{n=1}^{N-3}\frac{3}{2}\frac{[\hat{y}_{n+2} - \hat{y}_{n+1} + h(\gamma\hat{y}_n + \beta^{-2}x_n^3 - x_n)]^2}{h\sigma^2} + (N-3)\log\sigma^2\right]. \tag{4-3}$$

As for the linear Langevin system case, we begin by investigating the reliability of the estimators. One hundred simulated data sets are generated from the above Kramers oscillator with true parameters $\gamma = 0.5$, $\beta = 1/\sqrt{10}$, and $\sigma = 1$, and with initial condition $x_0 = y_0 = \frac{1}{2}$ and integration time interval $[0, 10^4]$. The estimators of (γ, β, σ) are computed for each data set. Empirical mean and standard deviation of the estimators are shown in Table 3. We observe that the biases in the estimators increase as h increases; in particular, the estimator of $\hat{\gamma}$ has a very large bias.

For the discrete-time approach, we have to select one of the four NARMA$(2, q)$ models, Models (M1)–(M4). We make the selection using data only from a single long trajectory (e.g., from the time interval $[0, T]$ with $T = 2^{18} \approx 2 \times 10^5$), and we use the first half of the data to estimate the parameters. We first estimate the

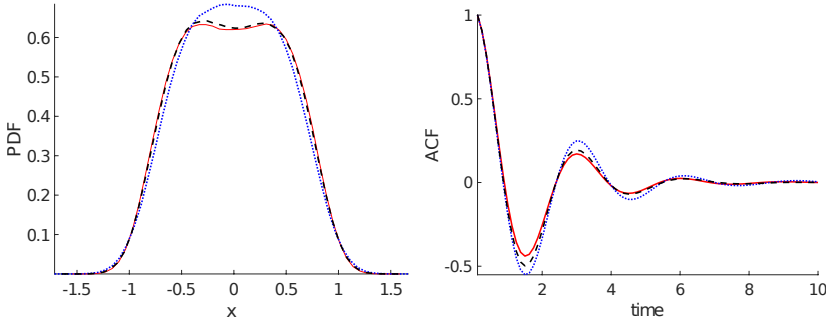


Figure 4. Empirical PDFs and ACFs of the NARMA models (M2) (blue dotted line), (M3) (black dashed line) and data (red solid line) in the case $h = \frac{1}{8}$. Model (M3) reproduces the ACF and PDF better than Model (M2).

Estimator	$h = \frac{1}{32}$	$h = \frac{1}{16}$	$h = \frac{1}{8}$
$\hat{a}_1$	1.9906 (0.0004)	1.9829 (0.0007)	1.9696 (0.0014)
$-\hat{a}_2$	0.9896 (0.0004)	0.9792 (0.0007)	0.9562 (0.0014)
$-\hat{b}_1$	0.3388 (0.1572)	0.6927 (0.0785)	1.2988 (0.0389)
$\hat{b}_2$	0.0300 (0.1572)	0.0864 (0.0785)	0.1462 (0.0386)
$\hat{b}_3$	0.0307 (0.1569)	0.0887 (0.0777)	0.1655 (0.0372)
$-\hat{\mu}$ $(\times 10^{-5})$	0.0377 (0.0000)	0.1478 (0.0000)	0.5469 (0.0001)
$\hat{\sigma}_W$	0.0045 (0.0000)	0.1119 (0.0001)	0.0012 (0.0000)

Table 4. Mean and standard deviation of the estimators of the parameters of the NARMA model (M3) with $q = 0$ in the discrete-time approach, computed from 100 simulations.

parameters for each NARMA model with $q = 0$ and $q = 1$, using the conditional likelihood method described in Section 3A. Then we make a selection by the criteria proposed in Section 3A. First, we test numerical stability by running the model for a large time for different realizations of the noise sequence. We find that for our model, using the values of h tested here, Model (M1) is often numerically unstable, so we do not compare it to the other schemes here. (In situations where the Euler scheme is more stable, e.g., for smaller values of h or for other models, we would expect it to be useful as the basis of a NARMA approximation.) Next, we test the performance of each of the models (M2)–(M4). The RMSEs of Models (M2) and (M3) with $q = 0$ and $q = 1$ and Model (M4) with $q = 1$ are shown in Figure 3. In the case $q = 1$, the RMSEs for Models (M2)–(M4) are very close, but they are larger than the RMSEs of Models (M2) and (M3) with $q = 0$. To make a further selection between Models (M2) and (M3) with $q = 0$, we test their reproduction of the long-term statistics. Figure 4 shows that Model (M3) reproduces the ACFs and PDFs better than Model (M2); hence, Model (M3) with $q = 0$ is selected.

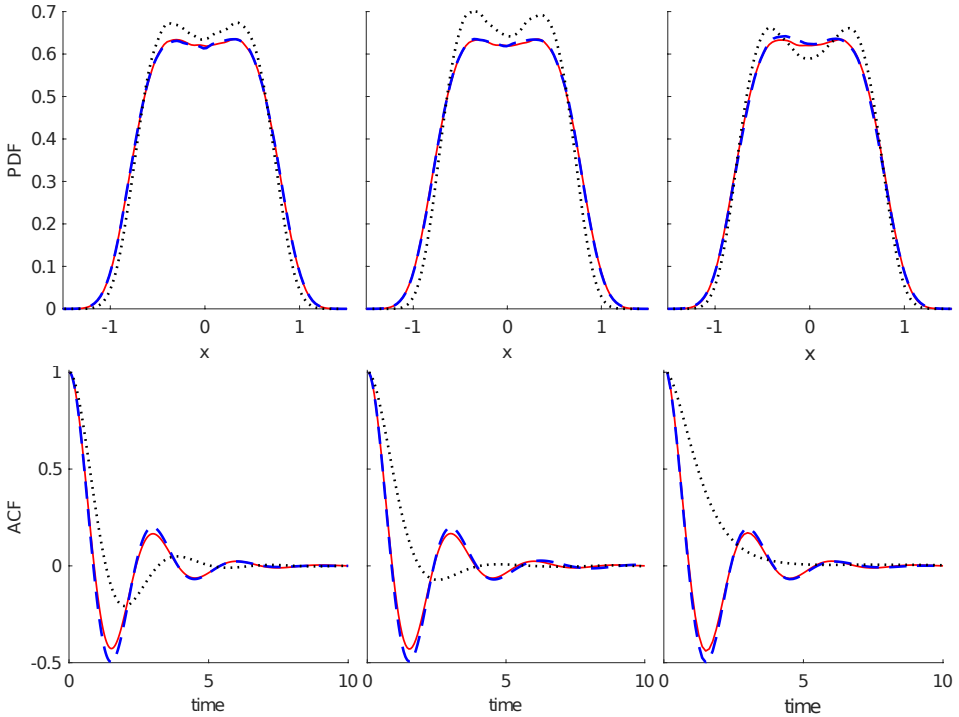


Figure 5. Empirical PDFs and ACFs of the NARMA model (M3) with $q = 0$ (blue dashed line) and the estimated Kramers system (black dotted line), in the cases $h = \frac{1}{32}$ (left), $h = \frac{1}{16}$ (center), and $h = \frac{1}{8}$ (right). These statistics are better reproduced (red solid line) by the NARMA models than by the estimated Kramers systems.

The mean and standard deviation of the estimated parameters of Model (M3) with $q = 0$ and 100 simulations are shown in Table 4. Unlike in the linear Langevin system case, we do not have a theoretical value for these parameters. However, note that when $h = \frac{1}{32}$, $\hat{a}_1$ and $\hat{a}_2$ are close to $2 - \gamma h + 0.5\gamma^2 h^2 = 1.9845$ and $-(1 - \gamma h + 0.5\gamma^2 h^2) = -0.9845$, respectively, which are the coefficients in (3-8) from the Itô–Taylor scheme. This indicates that when h is small, the NARMA model is close to the numerical scheme, because both the NARMA and the numerical scheme approximate the true system well. On the other hand, note that $\hat{\sigma}_W$ does not increase monotonically as h increases. This clearly distinguishes the NARMA model from the numerical schemes.

Next, we compare the performance of the NARMA model and the estimated Kramers system in reproducing long-term statistics and predicting short-term dynamics. The empirical PDFs and ACFs are shown in Figure 5. The NARMA models can reproduce the PDFs and ACFs equally well for three cases. The estimated Kramers system amplifies the depth of double wells in the PDFs, and it misses the oscillation of the ACFs.

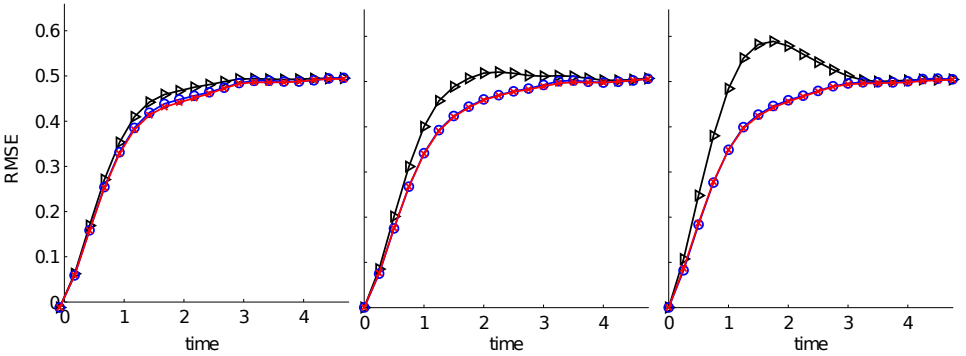


Figure 6. The Kramers system: RMSEs of 10^4 forecasting ensembles with size $N_{\text{ens}} = 20$, produced by the true Kramers system (black triangles), the Kramers system with estimated parameters (red x's), and the NARMA model (M3) (blue circles) with $q = 0$, in the cases $h = \frac{1}{32}$ (left), $h = \frac{1}{16}$ (center), and $h = \frac{1}{8}$ (right). The NARMA model has almost the same RMSEs as the true system for all the observation spacings, while the estimated system has larger RMSEs.

Results for RMSEs for $N_0 = 10^4$ ensembles with size $N_{\text{ens}} = 20$ are shown in Figure 6. The NARMA model reproduces almost exactly the RMSEs of the true Kramers system for all three step sizes, while the estimated Kramers system has increasing error as h increases, due to the increasing biases in the estimators.

Finally, in Figure 7, we show some results using a much smaller observation spacing, $h = \frac{1}{1024}$. Table 5 shows the estimated parameters, for both the continuous- and discrete-time models. (Here, the discrete-time model is (M2).) Consistent with the theory in [30], our parameter estimates for the continuous-time model are close to their true values for this small value of h. Figure 7 compares the RMSE of the continuous-time and discrete-time models on the same forecasting task as before. The continuous-time approach now performs much better, essentially as well as the true model. Even in this regime, however, the discrete-time approach remains competitive.

4C. ***Criteria for structure design.*** In the above structure selection between Models (M2) and (M3), we followed the criterion of selecting the one that fits the long-term statistics best. However, there is another practical criterion, namely whether the estimators converge as the number of samples increases. This is important because the estimators should converge to the true values of the parameters if the model is correct, due to the consistency discussed in Section 3A. Convergence can be tested by checking the oscillations of estimators as data length increases: if the oscillations are large, the estimators are likely not to converge, at least not quickly. Table 6 shows the estimators of the coefficients of the nonlinear terms in Models (M2) and (M3), for different lengths of data. The estimators $\hat{b}_1$, $\hat{b}_2$, and $\hat{b}_3$ of Model (M3) are

Continuous-time model parameters			Discrete-time model parameters		
$\hat{\gamma}$	$-\hat{\beta}$	$\hat{\sigma}$	$\hat{a}_1$	$-\hat{a}_2$	$-\hat{b}_1$
0.5163	0.3435	1.0006	1.9997	0.9997	0.0097
			$-\hat{b}_2$	$-\hat{\mu}$ $(\times 10^{-8})$	$\hat{\sigma_W}$ $(\times 10^{-10})$
			0.0169	2.0388	6.2165

Table 5. Estimated parameters for the continuous-time and discrete-time models.

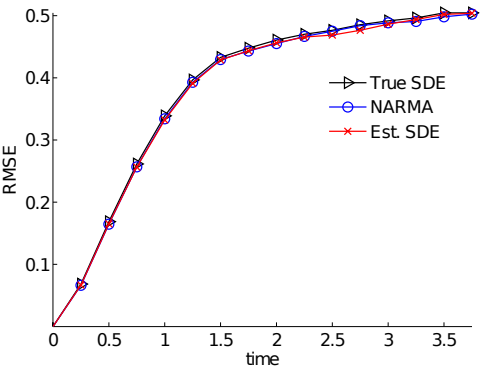


Figure 7. RMSEs of 10^3 forecasting ensembles with size $N_{\text{ens}} = 20$ with $h = \frac{1}{1024}$, produced by the true Kramers system (True SDE), the Kramers system with estimated parameters (Est. SDE), and the NARMA model (M2) with $q = 0$. Since $h = \frac{1}{1024}$ is relatively small, the NARMA model and the estimated system have almost the same RMSEs as the true system. Here the data is generated by the Itô–Taylor solver with step size $dt = 2^{-15} \approx 3 \times 10^{-5}$, and data length is $N = 2^{22} \approx 4 \times 10^6$.

Data length	Model (M2)		Model (M3)		
$(\times N)$	$-\hat{b}_1$	$-\hat{b}_2$	$-\hat{b}_1$	$\hat{b}_2$	$\hat{b}_3$
$\frac{1}{8}$	0.3090	0.3032	0.3622	0.0532	0.0563
$\frac{1}{4}$	0.3082	0.3049	0.3290	0.0208	0.0217
$\frac{1}{2}$	0.3088	0.3083	0.3956	0.0868	0.0845
1	0.3087	0.3054	0.3778	0.0691	0.0697

Table 6. Consistency test. Values of the estimators in the NARMA models (M2) and (M3) with $q = 0$. The data come from a long trajectory with observation spacing $h = \frac{1}{32}$. Here $N = 2^{22} \approx 4 \times 10^6$. As the length of data increases, the estimators of Model (M2) have much smaller oscillation than the estimators of Model (M3).

unlikely to be convergent, since they vary a lot for long data sets. On the contrary, the estimators $\hat{b}_1$ and $\hat{b}_2$ of Model (M2) have much smaller oscillations, and hence they are likely to be convergent.

Estimator	$h = \frac{1}{32}$	$h = \frac{1}{16}$	$h = \frac{1}{8}$
$\hat{a}_1$	1.9905 (0.0003)	1.9820 (0.0007)	1.9567 (0.0013)
$-\hat{a}_2$	0.9896 (0.0003)	0.9788 (0.0007)	0.9508 (0.0014)
$-\hat{b}_1$	0.3088 (0.0021)	0.6058 (0.0040)	1.1362 (0.0079)
$-\hat{b}_2$	0.3067 (0.0134)	0.5847 (0.0139)	0.9884 (0.0144)
$-\hat{\mu}$ $(\times 10^{-5})$	0.0340 (0.0000)	0.1193 (0.0000)	0.2620 (0.0001)
$\hat{\sigma}_W$	0.0045 (0.0000)	0.1119 (0.0001)	0.0012 (0.0000)

Table 7. Mean and standard deviation of the estimators of the parameters $(a_1, a_2, b_1, b_2, \mu, \sigma_W)$ of the NARMA model (M2) with $q = 0$ in the discrete-time approach, computed on 100 simulations.

These convergence tests agree with the statistics of the estimators on 100 simulations in Tables 4 and 7. Table 4 shows that the standard deviations of the estimators $\hat{b}_1$, $\hat{b}_2$, and $\hat{b}_3$ of Model (M3) are reduced by half as h doubles, which is the opposite of what is supposed to happen for an accurate model. On the contrary, Table 7 shows that the standard deviations of the parameters of Model (M2) increase as h doubles, as is supposed to happen for an accurate model.

In short, Model (M3) reproduces better long-term statistics than Model (M2), but the estimators of Model (M2) are statistically better (e.g., in rate of convergence) than the estimators of Model (M3). However, the two have almost the same prediction skill as shown in Figure 3, and both are much better than the continuous-time approach. It is unclear which model approximates the true process better, and it is likely that neither of them is optimal. Also, it is unclear which criterion is better for structure selection: fitting the long-term statistics or consistency of estimators. We leave these issues to be addressed in future work.

5. Concluding discussion

We have compared a discrete-time approach and a continuous-time approach to the data-based stochastic parametrization of a dynamical system, in a situation where the data are known to have been generated by a hypoelliptic stochastic system of a given form. In the continuous-time case, we first estimated the coefficients in the given equations using the data, and then solved the resulting differential equations; in the discrete-time model, we chose structures with terms suggested by numerical algorithms for solving the equations of the given form, with coefficients estimated using the data.

As discussed in our earlier papers [8; 21], the discrete-time approach has several a priori advantages:

- The inverse problem of estimating the parameters in a model from discrete data is in general better-posed in a discrete-time than in a continuous-time model.

In particular, the discrete-time representation is more tolerant of relatively large observation spacings.

- Once the discrete-time parametrization has been derived, it can be used directly in numerical computation; there is no need of further approximation. This is not a major issue in the present paper where the equations are relatively simple, but we expect it to grow in significance as the size of problems increases.

Our example validates the first of these points; the discrete-time approximations generally have better prediction skills than the continuous-time parametrization, especially when the observation spacing is relatively large. This was also the main source of error in the continuous models discussed in [8]; note that the method for parameter estimation in that earlier paper was completely different. Our discrete-time models also have better numerical properties; e.g., when all else is equal, they are more stable and produce more accurate long-term statistics than their continuous-time counterparts.

We expect the advantages of the discrete-time approach to become more marked as one proceeds to analyze systems of growing complexity, particularly larger, more chaotic dynamical systems. A number of questions remain, first and foremost being the identification of effective structures; this is of course a special case of the difficulty in identifying effective bases in the statistical modeling of complex phenomena. In the present paper we introduced the idea of using terms derived from numerical approximations; different ideas were introduced in our earlier work [21]. More work is needed to generate general tools for structure determination.

Another challenge is that, even when one has derived a small number of potential structures, we currently do not have a systematic way to identify the most effective model. Thus, the selection of a suitable discrete-time model can be labor-intensive, especially compared to the continuous-time approach in situations where a parametric family containing the true model (or a good approximation thereof) is known. On the other hand, continuous-time approaches, in situations where no good family of models is known, would face similar difficulties.

Finally, another open question is whether discrete-time approaches generally produce more accurate predictions than continuous-time approaches for strongly chaotic systems. Previous work has suggested that the answer may be yes. We plan to address this question more systematically in future work.

Acknowledgments

We would like to thank the anonymous referee and Dr. Robert Saye for helpful suggestions. Lin is supported in part by the National Science Foundation under grant DMS-1418775. Chorin and Lu are supported in part by the Director, Office of Science, Computational and Technology Research, U.S. Department of Energy, under

Contract Number DE-AC02-05CH11231, and by the National Science Foundation under grant DMS-1419044.

Appendix A: Solutions to the linear Langevin equation

Denoting

$$\boldsymbol{X}_t = \begin{pmatrix} x_t \\ y_t \end{pmatrix}, \qquad \boldsymbol{A} = \begin{pmatrix} 0 & 1 \\ -\alpha & -\gamma \end{pmatrix}, \qquad \boldsymbol{e} = \begin{pmatrix} 0 \\ \sigma \end{pmatrix},$$

we can write (3-4) as

$$d\boldsymbol{X}_t = \boldsymbol{A}\boldsymbol{X}_t\, dt + \boldsymbol{e}\, dB_t.$$

Its solution is

$$\boldsymbol{X}_t = e^{\boldsymbol{A}t}\boldsymbol{X}_0 + \int_0^t e^{\boldsymbol{A}(t-u)}\boldsymbol{e}\, dB_u.$$

The solution at discrete times can be written as

$$x_{(n+1)h} = a_{11}x_{nh} + a_{12}y_{nh} + W_{n+1,1},$$
$$y_{(n+1)h} = a_{21}x_{nh} + a_{22}y_{nh} + W_{n+1,2},$$

where $a_{ij} = (e^{\boldsymbol{A}h})_{ij}$ for $i, j = 1, 2$, and

$$W_{n+1,i} = \sigma \int_0^h a_{i2}(u)\, dB(nh + u) \tag{A-1}$$

with $a_{i2}(u) = (e^{\boldsymbol{A}(h-u)})_{i2}$ for $i = 1, 2$. Note that if $a_{12} \neq 0$, then from the first equation we get $y_{nh} = (x_{(n+1)h} - a_{11}x_{nh} - V_{n+1,1})/a_{12}$. Substituting it into the second equation we obtain

$$x_{(n+2)h} = (a_{11}+a_{22})x_{(n+1)h} + (a_{12}a_{21} - a_{11}a_{22})x_{nh} - a_{22}W_{n+1,1} + a_{12}W_{n+1,2} + W_{n+2,1}.$$

Combining with the fact that $a_{11} + a_{22} = \operatorname{trace}(e^{\boldsymbol{A}h})$ and $a_{12}a_{21} - a_{11}a_{22} = -e^{-\gamma h}$, we have

$$x_{(n+2)h} = \operatorname{trace}(e^{\boldsymbol{A}h})x_{(n+1)h} - e^{-\gamma h}x_{nh} - a_{22}W_{n+1,1} + W_{n+2,1} + a_{12}W_{n+1,2}. \tag{A-2}$$

Clearly, the process $\{x_{nh}\}$ is a centered Gaussian process, and its distribution is determined by its autocovariance function. Conditionally on $\boldsymbol{X}_0$, the distribution of $\boldsymbol{X}_t$ is $\mathcal{N}(e^{\boldsymbol{A}t}\boldsymbol{X}_0, \boldsymbol{\Sigma}(t))$, where $\boldsymbol{\Sigma}(t) := \int_0^t e^{\boldsymbol{A}u}\boldsymbol{e}\boldsymbol{e}^T e^{\boldsymbol{A}^T u}\, du$. Since $\alpha, \gamma > 0$, the real parts of the eigenvalues of $\boldsymbol{A}$, denoted by λ_1 and λ_2, are negative. The stationary distribution is $\mathcal{N}(0, \boldsymbol{\Sigma}(\infty))$, where $\boldsymbol{\Sigma}(\infty) = \lim_{t\to\infty} \boldsymbol{\Sigma}(t)$. If $\boldsymbol{X}_0$ has distribution $\mathcal{N}(0, \boldsymbol{\Sigma}(\infty))$, then the process $(\boldsymbol{X}_t)$ is stationary, and so is the observed process $\{x_{nh}\}$. The following lemma computes the autocorrelation function of the stationary process $\{x_{nh}\}$.

Lemma A.1. *Assume that the system (3-4) is stationary. Denote by* $\{\gamma_j\}_{j=1}^{\infty}$ *the autocovariance function of the stationary process* $\{x_{nh}\}$; *i.e.,* $\gamma_j := \mathbb{E}[x_{kh}x_{(k+j)h}]$ *for* $j \geq 0$. *Then* $\gamma_0 = \sigma^2/(2\alpha\gamma)$, *and* γ_j *can be represented as*

$$\gamma_j = \gamma_0 \times \begin{cases} \dfrac{1}{\lambda_1 - \lambda_2}(\lambda_1 e^{\lambda_2 jh} - \lambda_2 e^{\lambda_1 jh}) & \text{if } \gamma^2 - 4\alpha \neq 0, \\ e^{\lambda_0 jh}(1 - \lambda_0 jh) & \text{if } \gamma^2 - 4\alpha = 0 \end{cases}$$

for all $j \geq 0$, *where* λ_1 *and* λ_2 *are the different solutions to* $\lambda^2 + \gamma\lambda + \alpha = 0$ *when* $\gamma^2 - 4\alpha \neq 0$, *and* $\lambda_0 = -\gamma/2$.

Proof. Let $\mathbf{\Gamma}(j) := \mathbb{E}[\boldsymbol{X}_{kh}\boldsymbol{X}_{(k+j)h}^T] = \mathbf{\Sigma}(\infty)e^{\boldsymbol{A}^T jh}$ for $j \geq 0$. Note that $\gamma_j = \mathbf{\Gamma}_{11}(j)$, i.e., γ_j is the first element of the matrix $\mathbf{\Gamma}(j)$. Then it follows that

$$\gamma_0 = \mathbf{\Sigma}_{11}(\infty), \qquad \gamma_j = (\mathbf{\Sigma}(\infty)e^{\boldsymbol{A}^T jh})_{11}.$$

If $\gamma^2 - 4\alpha \neq 0$, then $\boldsymbol{A}$ has two different eigenvalues λ_1 and λ_2 and can be written as

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{-1} \quad \text{with } \boldsymbol{Q} = \begin{pmatrix} 1 & 1 \\ \lambda_1 & \lambda_2 \end{pmatrix} \text{ and } \boldsymbol{\Lambda} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}.$$

The covariance matrix $\mathbf{\Sigma}(\infty)$ can be computed as

$$\mathbf{\Sigma}(\infty) = \lim_{t\to\infty} \int_0^t \boldsymbol{Q}e^{\boldsymbol{\Lambda}u}\boldsymbol{Q}^{-1}\boldsymbol{e}\boldsymbol{e}^T\boldsymbol{Q}^{-T}e^{\boldsymbol{\Lambda}^T u}\boldsymbol{Q}^T du = \sigma^2 \begin{pmatrix} 1/(2ab) & 0 \\ 0 & -1/(2b) \end{pmatrix}. \quad \text{(A-3)}$$

This gives $\gamma_0 = \mathbf{\Sigma}_{11}(\infty) = \sigma^2/(2\gamma\alpha)$ and for $j > 0$,

$$\gamma_j = \mathbf{\Sigma}_{11}(\infty)(e^{\boldsymbol{A}^T jh})_{11} = \frac{1}{\lambda_1 - \lambda_2}(\lambda_1 e^{\lambda_2 jh} - \lambda_2 e^{\lambda_1 jh})\gamma(0).$$

In the case $\gamma^2 - 4\alpha = 0$, $\boldsymbol{A}$ has a single eigenvalue $\lambda_0 = -\gamma/2$, and it can be transformed to a Jordan block

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{-1} \quad \text{with } \boldsymbol{Q} = \begin{pmatrix} 1 & 0 \\ \lambda_0 & 1 \end{pmatrix} \text{ and } \boldsymbol{\Lambda} = \begin{pmatrix} \lambda_0 & 1 \\ 0 & \lambda_0 \end{pmatrix}.$$

This leads to the same $\mathbf{\Sigma}(\infty)$ as in (A-3). Similarly, we have $\gamma_0 = \sigma^2/(2\gamma\alpha)$ and

$$\gamma_j = \mathbf{\Sigma}_{11}(\infty)(e^{\boldsymbol{A}^T jh})_{11} = e^{\lambda_0 jh}(1 - \lambda_0 jh)\gamma_0. \qquad \square$$

Appendix B: ARMA processes

We review the definition and computation of the autocovariance function of ARMA processes in this subsection. For more details, we refer to [6, §3.3].

Definition B.1. The process $\{X_n : n \in \mathbb{Z}\}$ is said to be an ARMA(p, q) process if it is a stationary process satisfying

$$X_n - \phi_1 X_{n-1} - \cdots - \phi_p X_{n-p} = W_n + \theta_1 W_{n-1} + \cdots + \theta_q W_{n-q}, \tag{B-1}$$

for every n, where the $\{W_n\}$ are i.i.d. $\mathcal{N}(0, \sigma_W^2)$, and if the polynomials $\phi(z) := 1 - \phi_1 z - \cdots - \phi_p z^p$ and $\theta(z) := 1 + \theta_1 z + \cdots + \theta_q z^q$ have no common factors. If $\{X_n - \mu\}$ is an ARMA(p, q) process, then $\{X_n\}$ is said to be an ARMA(p, q) process with mean μ. The process is causal if $\phi(z) \neq 0$ for all $|z| \leq 1$. The process is invertible if $\theta(z) \neq 0$ for all $|z| \leq 1$.

The autocovariance function $\{\gamma(k)\}_{k=1}^{\infty}$ of an ARMA(p, q) can be computed from the following difference equations, which are obtained by multiplying each side of (B-1) by X_{n-k} and taking expectations:

$$\gamma(k) - \phi_1 \gamma(k-1) - \cdots - \phi_p \gamma(k-p) = \sigma_W^2 \sum_{k \leq j \leq q} \theta_j \psi_{j-k},$$

$$0 \leq k < \max\{p, q+1\}, \tag{B-2}$$

$$\gamma(k) - \phi_1 \gamma(k-1) - \cdots - \phi_p \gamma(k-p) = 0, \qquad k \geq \max\{p, q+1\}, \tag{B-3}$$

where ψ_j in (B-2) is computed as (letting $\theta_0 := 1$ and $\theta_j = 0$ if $j > q$)

$$\psi_j = \begin{cases} \theta_j + \sum_{0<k\leq j} \phi_k \psi_{j-k} & \text{for } j < \max\{p, q+1\}, \\ \sum_{0<k\leq p} \phi_k \psi_{j-k} & \text{for } j \geq \max\{p, q+1\}. \end{cases}$$

Denote by $\{\zeta_i : i = 1, \ldots, k\}$ the distinct zeros of $\phi(z) := 1 - \phi_1 z - \cdots - \phi_p z^p$, and let r_i be the multiplicity of ζ_i (hence $\sum_{i=1}^{k} r_i = p$). The general solution of the difference (B-3) is

$$\gamma(n) = \sum_{i=1}^{k} \sum_{j=0}^{r_i - 1} \beta_{ij} n^j \zeta_i^{-n} \quad \text{for } n \geq \max\{p, q+1\} - p, \tag{B-4}$$

where the p constants β_{ij} (hence the values of $\gamma(j)$ for $0 \leq j < \max\{p, q+1\} - p$) are determined from (B-2).

Example B.2 (ARMA$(2, 0)$). The autocovariance function for an ARMA$(2, 0)$ process $X_n - \phi_1 X_{n-1} - \phi_2 X_{n-2} = W_n$ is

$$\gamma(n) = \begin{cases} \beta_1 \zeta_1^{-n} + \beta_2 \zeta_2^{-n} & \text{if } \phi_1^2 + 4\phi_2 \neq 0, \\ (\beta_1 + \beta_2 n)\zeta^{-n} & \text{if } \phi_1^2 + 4\phi_2 = 0 \end{cases}$$

for $n \geq 0$, where ζ_1, ζ_2, or ζ are the zeros of $\phi(z) = 1 - \phi_1 z - \phi_2 z^2$. The constants β_1 and β_2 are computed from the equations

$$\gamma(0) - \phi_1 \gamma(1) - \phi_2 \gamma(2) = \sigma_W^2,$$
$$\gamma(1) - \phi_1 \gamma(0) - \phi_2 \gamma(1) = 0.$$

Example B.3 (ARMA(2, 1)). We have $\psi_0 = 1$ and $\psi_1 = \phi_1$ for an ARMA(2, 1) process $X_n - \phi_1 X_{n-1} - \phi_2 X_{n-2} = W_n + \theta_1 W_{n-1}$. Its autocovariance function is of the same form as that in Example B.2, where the constants β_1 and β_2 are computed from the equations

$$\gamma(0) - \phi_1\gamma(1) - \phi_2\gamma(2) = \sigma_W^2(1 + \theta_1^2 + \theta_1\phi_1),$$
$$\gamma(1) - \phi_1\gamma(0) - \phi_2\gamma(1) = \sigma_W^2\theta_1.$$

Appendix C: Numerical schemes for hypoelliptic SDEs with additive noise

Here we briefly review the two numerical schemes, the Euler–Maruyama scheme and the Itô–Taylor scheme of strong order 2.0, for hypoelliptic systems with additive noise

$$dx = y\,dt,$$
$$dy = a(x, y)\,dt + \sigma\,dB_t,$$

where $a : \mathbb{R}^2 \to \mathbb{R}$ satisfies suitable conditions so that the system is ergodic.

In the following, the step size of all schemes is h, and $W_n = \sigma\sqrt{h}\xi_n$ and $Z_n = \sigma h^{3/2}(\xi_n + \eta_n/\sqrt{3})$, where $\{\xi_n\}$ and $\{\eta_n\}$ are two i.i.d. sequences of $\mathcal{N}(0, 1)$ random variables.

Scheme C.1 (Euler–Maruyama).
$$x_{n+1} = x_n + y_n h,$$
$$y_{n+1} = y_n + ha(x_n, y_n) + W_{n+1}.$$

Scheme C.2 (Itô–Taylor scheme of strong order 2.0).

$$x_{n+1} = x_n + hy_n + 0.5h^2 a(x_n, y_n) + Z_{n+1},$$
$$y_{n+1} = y_n + ha(x_n, y_n) + 0.5h^2[a_x(x_n, y_n)y_n + (aa_y + 0.5\sigma^2 a_{yy})(x_n, y_n)] + W_{n+1} + a_y(x_n, y_n)Z_{n+1} + a_{yy}(x_n, y_n)\sigma^2\tfrac{1}{6}h(W_{n+1}^2 - h).$$

The Itô–Taylor scheme of order 2.0 can be derived as follows (see, e.g., works of Kloeden and Platen [14; 18]). The differential equation can be rewritten in the integral form

$$x_t = x_{t_0} + \int_{t_0}^{t} y_s\,ds,$$
$$y_t = y_{t_0} + \int_{t_0}^{t} a(x_s, y_s)\,ds + \sigma(B_t - B_{t_0}).$$

We start from the Itô–Taylor expansion of x:

$$x_{t_{n+1}} = x_{t_n} + hy_{t_n} + \int_{t_n}^{t_{n+1}}\int_{t_n}^{t} a(x_s, y_s)\,ds\,dt + \sigma I_{10}^{n+1}$$
$$= x_{t_n} + hy_{t_n} + 0.5h^2 a(x_{t_n}, y_{t_n}) + \sigma I_{10}^{n+1} + O(h^{5/2}),$$

where $I_{10}^{n+1} := \int_{t_n}^{t_{n+1}} (B_t - B_{t_n})\, dt$. To get a higher-order scheme for y, we apply Itô's chain rule to $a(x_t, y_t)$:

$$a(x_t, y_t) = a(x_s, y_s) + \int_s^t [a_x(x_r, y_r) y_r + (a a_y + 0.5\sigma^2 a_{yy})(x_r, y_r)]\, dr + \sigma \int_s^t a_y(x_r, y_r)\, dB_r.$$

This leads to the Itô–Taylor expansion for y (up to the order 2.0):

$$\begin{aligned} y_{t_{n+1}} &= y_{t_n} + \int_{t_n}^{t_{n+1}} a(x_s, y_s)\, ds + \sigma(B_{t_{n+1}} - B_{t_n}) \\ &= y_{t_n} + h a(x_{t_n}, y_{t_n}) + \sigma(B_{t_{n+1}} - B_{t_n}) + a_y(x_{t_n}, y_{t_n}) \sigma I_{10}^{n+1} + a_{yy}(x_{t_n}, y_{t_n}) \sigma^2 I_{110}^{n+1} \\ &\quad + 0.5h^2 [a_x(x_{t_n}, y_{t_n}) y_{t_n} + (a a_y + 0.5\sigma^2 a_{yy})(x_{t_n}, y_{t_n})] + O(h^{5/2}), \end{aligned}$$

where $I_{110}^{n+1} = \int_{t_n}^{t_{n+1}} \int_{t_n}^{t} (B_s - B_{t_n})\, dB_s\, dt$. Representing $\sigma(B_{t_{n+1}} - B_{t_n})$, σI_{10}^{n+1}, and I_{110}^{n+1} by W_{n+1}, Z_{n+1}, and $\frac{1}{6}h(W_{n+1}^2 - h)$, respectively, we obtain Scheme C.2.

References

[1] E. B. Andersen, *Asymptotic properties of conditional maximum-likelihood estimators*, J. Roy. Statist. Soc. (B) **32** (1970), 283–301.

[2] D. F. Anderson and J. C. Mattingly, *A weak trapezoidal method for a class of stochastic differential equations*, Commun. Math. Sci. **9** (2011), no. 1, 301–318.

[3] L. Arnold and P. Imkeller, *The Kramers oscillator revisited*, Stochastic processes in physics, chemistry, and biology (J. A. Freund and T. Pöschel, eds.), Lecture Notes in Physics, no. 557, Springer, Berlin, 2000, pp. 280–291.

[4] P. J. Brockwell, *Continuous-time ARMA processes*, Stochastic processes: theory and methods (V. N. Gudivada, V. V. Raghavan, V. Govindaraju, and C. R. Rao, eds.), Handbook of Statist., no. 19, North-Holland, Amsterdam, 2001, pp. 249–276.

[5] _____, *Recent results in the theory and applications of CARMA processes*, Ann. Inst. Statist. Math. **66** (2014), no. 4, 647–685.

[6] P. J. Brockwell and R. A. Davis, *Time series: theory and methods*, 2nd ed., Springer, New York, 1991.

[7] P. J. Brockwell, R. A. Davis, and Y. Yang, *Continuous-time Gaussian autoregression*, Stat. Sinica **17** (2007), no. 1, 63–80.

[8] A. J. Chorin and F. Lu, *Discrete approach to stochastic parametrization and dimensional reduction in nonlinear dynamics*, P. Natl. Acad. Sci. USA **112** (2015), no. 32, 9804–9809.

[9] S. Ditlevsen and M. Sørensen, *Inference for observations of integrated diffusion processes*, Scand. J. Statist. **31** (2004), no. 3, 417–429.

[10] D. Frenkel and B. Smit, *Understanding molecular simulation: from algorithms to applications*, 2nd ed., Computational Science, no. 1, Academic, San Diego, 2002.

[11] A. Gloter, *Parameter estimation for a discretely observed integrated diffusion process*, Scand. J. Statist. **33** (2006), no. 1, 83–104.

[12] G. A. Gottwald, D. T. Crommelin, and C. L. E. Franzke, *Stochastic climate theory*, Nonlinear climate dynamics (C. L. E. Franzke and T. J. O'Kane, eds.), Cambridge University, 2016.

[13] J. D. Hamilton, *Time series analysis*, Princeton University, 1994.

[14] Y. Hu, *Strong and weak order of time discretization schemes of stochastic differential equations*, Séminaire de Probabilités, XXX (J. Azéma, M. Emery, and M. Yor, eds.), Lecture Notes in Math., no. 1626, Springer, Berlin, 1996, pp. 218–227.

[15] G. Hummer, *Position-dependent diffusion coefficients and free energies from Bayesian analysis of equilibrium and replica molecular dynamics simulations*, New J. Phys. **7** (2005), no. 34.

[16] A. C. Jensen, *Statistical inference for partially observed diffusion processes*, Ph.D. thesis, University of Copenhagen, 2014.

[17] R. H. Jones, *Fitting a continuous time autoregressive to discrete data*, Applied time series analysis, II: Proceedings of the second Applied Time Series Symposium (D. F. Findley, ed.), Academic, New York, 1981, pp. 651–682.

[18] P. E. Kloeden and E. Platen, *Numerical solution of stochastic differential equations*, Applications of Math., no. 23, Springer, Berlin, 1992.

[19] D. Kondrashov, M. D. Chekroun, and M. Ghil, *Data-driven non-Markovian closure models*, Phys. D **297** (2015), 33–55.

[20] H. A. Kramers, *Brownian motion in a field of force and the diffusion model of chemical reactions*, Physica **7** (1940), 284–304.

[21] F. Lu, K. K. Lin, and A. J. Chorin, *Data-based stochastic model reduction for the Kuramoto–Sivashinsky equation*, Phys. D **340** (2017), 46–57.

[22] A. J. Majda and J. Harlim, *Physics constrained nonlinear regression models for time series*, Nonlinearity **26** (2013), no. 1, 201–217.

[23] J. C. Mattingly, A. M. Stuart, and D. J. Higham, *Ergodicity for SDEs and approximations: locally Lipschitz vector fields and degenerate noise*, Stochastic Process. Appl. **101** (2002), no. 2, 185–232.

[24] J. C. Mattingly, A. M. Stuart, and M. V. Tretyakov, *Convergence of numerical time-averaging and stationary measures via Poisson equations*, SIAM J. Numer. Anal. **48** (2010), no. 2, 552–577.

[25] G. N. Milstein and M. V. Tretyakov, *Computing ergodic limits for Langevin equations*, Phys. D **229** (2007), no. 1, 81–95.

[26] D. Nualart, *The Malliavin calculus and related topics*, 2nd ed., Springer, Berlin, 2006.

[27] A. W. Phillips, *The estimation of parameters in systems of stochastic differential equations*, Biometrika **46** (1959), no. 1–2, 67–76.

[28] Y. Pokern, A. M. Stuart, and P. Wiberg, *Parameter estimation for partially observed hypoelliptic diffusions*, J. Roy. Statist. Soc. (B) **71** (2009), no. 1, 49–73.

[29] B. L. S. Prakasa Rao, *Statistical inference for diffusion type processes*, Kendall's Library of Statistics, no. 8, Arnold, London, 1999.

[30] A. Samson and M. Thieullen, *A contrast estimator for completely or partially observed hypoelliptic diffusion*, Stochastic Process. Appl. **122** (2012), no. 7, 2521–2552.

[31] L. Schimansky-Geier and H. Herzel, *Positive Lyapunov exponents in the Kramers oscillator*, J. Stat. Phys. **70** (1993), no. 1–2, 141–147.

[32] H. Sørensen, *Parametric inference for diffusion processes observed at discrete points in time: a survey*, Int. Stat. Rev. **72** (2004), no. 3, 337–354.

[33] M. Sørensen, *Estimating functions for diffusion-type processes*, Statistical methods for stochastic differential equations (M. Kessler, A. Lindner, and M. Sørensen, eds.), Monogr. Statist. Appl. Probab., no. 124, CRC, Boca Raton, FL, 2012, pp. 1–107.

[34] D. Talay, *Stochastic Hamiltonian systems: exponential convergence to the invariant measure, and discretization by the implicit Euler scheme*, Markov Process. Related Fields **8** (2002), no. 2, 163–198.

Received May 31, 2016. Revised December 6, 2016.

FEI LU: feilu@berkeley.edu
Department of Mathematics, University of California, Berkeley, Evans Hall, Berkeley, CA 94720-3840, United States

and

Lawrence Berkeley National Laboratory, Berkeley, CA 94720, United States

KEVIN K. LIN: klin@math.arizona.edu
Department of Mathematics, University of Arizona, 617 North Santa Rita Avenue, Tucson, AZ 85721-0089, United States

ALEXANDRE J. CHORIN: chorin@math.berkeley.edu
Department of Mathematics, University of California, Berkeley, Evans Hall, Berkeley, CA 94720-3840, United States

and

Lawrence Berkeley National Laboratory, Berkeley, CA 94720, United States

COMM. APP. MATH. AND COMP. SCI.
Vol. 11, No. 2, 2016
dx.doi.org/10.2140/camcos.2016.11.217

HYDRODYNAMICS OF SUSPENSIONS OF PASSIVE AND ACTIVE RIGID PARTICLES: A RIGID MULTIBLOB APPROACH

FLORENCIO BALBOA USABIAGA, BAKYTZHAN KALLEMOV, BLAISE DELMOTTE, AMNEET PAL SINGH BHALLA, BOYCE E. GRIFFITH AND ALEKSANDAR DONEV

We develop a rigid multiblob method for numerically solving the mobility problem for suspensions of passive and active rigid particles of complex shape in Stokes flow in unconfined, partially confined, and fully confined geometries. As in a number of existing methods, we discretize rigid bodies using a collection of minimally resolved spherical blobs constrained to move as a rigid body, to arrive at a potentially large linear system of equations for the unknown Lagrange multipliers and rigid-body motions. Here we develop a block-diagonal preconditioner for this linear system and show that a standard Krylov solver converges in a modest number of iterations that is essentially independent of the number of particles. Key to the efficiency of the method is a technique for fast computation of the product of the blob-blob mobility matrix and a vector. For unbounded suspensions, we rely on existing analytical expressions for the Rotne–Prager–Yamakawa tensor combined with a fast multipole method (FMM) to obtain linear scaling in the number of particles. For suspensions sedimented against a single no-slip boundary, we use a direct summation on a graphical processing unit (GPU), which gives quadratic asymptotic scaling with the number of particles. For fully confined domains, such as periodic suspensions or suspensions confined in slit and square channels, we extend a recently developed rigid-body immersed boundary method by B. Kallemov, A. P. S. Bhalla, B. E. Griffith, and A. Donev (Commun. Appl. Math. Comput. Sci. **11** (2016), no. 1, 79–141) to suspensions of freely moving passive or active rigid particles at zero Reynolds number. We demonstrate that the iterative solver for the coupled fluid and rigid-body equations converges in a bounded number of iterations regardless of the system size. In our approach, each iteration only requires a few cycles of a geometric multigrid solver for the Poisson equation, and an application of the block-diagonal preconditioner, leading to linear scaling with the number of particles. We optimize a number of parameters in the iterative solvers and apply our method to a variety of benchmark problems to carefully assess the accuracy of the rigid multiblob approach as a function of the resolution. We also model the dynamics of colloidal particles studied in recent experiments, such as passive boomerangs in a slit channel, as well as a pair of non-Brownian active nanorods sedimented against a wall.

MSC2010: 76M25.
Keywords: Stokes flow, colloidal suspensions, Stokesian dynamics, immersed boundary method.

1. Introduction

The study of the hydrodynamics of colloidal suspensions of passive particles is an old yet still active subject in soft condensed matter physics and chemical engineering. In recent years there has been a growing interest in suspensions of active colloids [79], which exhibit rich collective behaviors quite distinct from those of passive suspensions. There is a growing number of computational methods for modeling active suspensions [50; 93; 32; 125; 106; 70; 119; 118], which are typically built upon well-developed techniques for passive suspensions in steady Stokes flow, i.e., at zero Reynolds number. Since active particles typically have metallic subcomponents, they are often significantly denser than the solvent and sediment toward the bottom wall, making it necessary to address confinement and implement nonperiodic boundary conditions in any method aimed at simulating experimentally relevant configurations. Furthermore, since collective motions seen in active suspensions involve large numbers of particles, and since hydrodynamic interactions among particles decay slowly like the inverse of the distance, it is crucial to develop methods that can capture long-ranged hydrodynamic effects, yet still scale to tens or hundreds of thousands of particles.

For suspensions of passive particles the methods of Brownian [68; 58] and Stokesian dynamics [117; 122] have dominated in chemical engineering, and related techniques have been used in biochemical engineering [62; 25; 104; 48; 35]. These methods simulate the overdamped (diffusive) dynamics of the particles by using Green's functions for steady Stokes flow to capture the effect of the fluid. While this sort of implicit solvent approach works very well in many situations, it has several notable technical difficulties: achieving near-linear scaling for many-particle systems is technically challenging, handling nontrivial boundary conditions (bounded systems) is complicated and has to be done on a case-by-case basis [69; 122; 123; 124; 86; 2; 88; 90; 96; 77; 57], generalizations to nonspherical (and in particular complex) particle shapes is difficult, and including thermal fluctuations is nontrivial due to the need to obtain stochastic increments with the desired covariance. In this work we develop relatively low-accuracy but flexible and simple rigid multiblob methods that address these difficulties. Our approach builds heavily on a number of existing techniques, combining elements from several distinct but related methods. We extensively test the proposed methods and study their accuracy and performance on a number of test problems.

The continuum formulation of the Stokes equations with suitable boundary conditions on the surfaces of a collection of rigid particles is well-known and summarized in more detail in Appendix A. Due to the linearity of the Stokes equations, there is an affine mapping from the applied forces $\boldsymbol{f}$ and torques $\boldsymbol{\tau}$ and any specified *apparent slip* velocity due to active boundary layers $\breve{\boldsymbol{u}}$, to the

resulting particle motion given by the linear velocities $\boldsymbol{u}$ and the angular velocities $\boldsymbol{\omega}$. Specifically,

$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{\omega} \end{bmatrix} = \mathcal{N} \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{\tau} \end{bmatrix} - \breve{\mathcal{N}}\, \breve{\boldsymbol{u}}, \tag{1}$$

where $\mathcal{N}$ is the *mobility matrix*, and $\breve{\mathcal{N}}$ is an *active mobility* linear operator. The *mobility problem* consists of computing the rigid-body motion given the applied forces and torques and apparent slip. The inverse of this problem is the *resistance problem*, which computes the forces and torques on the body given a specified motion of the body and active slip. Solving the mobility problem is a key component of any numerical method for modeling the deterministic or fluctuating (Brownian) dynamics of the particles.

In this paper we develop a *mobility solver* for suspensions of rigid particles immersed in viscous fluid; specifically, we develop novel preconditioners for iterative solvers for the unknown motions of the rigid bodies, given the applied external forces and torques as well as active apparent slip on the surface of the particles. As we discuss in more detail in the body of the paper, our formulation can readily solve the resistance problem; however, our iterative solvers will prove to be more scalable for mobility computations (which are of primary interest) than for resistance computations. Key to the success of our iterative solvers is the idea that instead of eliminating variables using *exact* Schur complements and solving a *reduced* system iteratively, as done in the majority of existing methods [126; 125; 31], one should instead iteratively solve an *extended* system that includes all of the variables. This has the key advantage that the matrix-vector product becomes an efficient direct calculation, and the Schur complement can be computed only *approximately* and used to construct an effective preconditioner.

Like many other authors, we construct rigid bodies of essentially arbitrary shape as a collection of rigidly connected collection of "blobs" or "beads" forming a composite object [126] that we will refer to as a *rigid multiblob*. The hydrodynamic interactions between blobs are represented using a Rotne–Prager tensor generalized to the desired domain geometry (boundary conditions) [130]; specifically, we use the Rotne–Prager–Yamakawa (RPY) tensor [113] for an unbounded domain, and the Rotne–Prager–Blake (RPB) tensor [122] for a half-space domain. In Section 2 we describe how to obtain the hydrodynamic coupling between a large collection of rigid multiblobs by solving a large linear system for Lagrange multipliers enforcing the rigidity. A key contribution of our work is to develop an indefinite saddle-point preconditioner for iterative solution of the resulting linear system. This preconditioner is based on a block-diagonal approximation of the blob-blob mobility matrix, in which all hydrodynamic interactions among distinct bodies (more precisely, among blobs on distinct bodies) are neglected. The only system-specific component

is the implementation of a fast matrix-vector multiplication routine, which in turn requires a scalable method for multiplying the RPY mobility matrix by a vector.

For simple geometries such as an unbounded domain or particles sedimented next to a no-slip boundary, simple analytical formulas for the RPY tensor are well-known [122; 130], and can be used to construct an efficient matrix-vector multiplication routine, for example, using fast multipole methods (FMMs) [89; 51], or even direct summation on a GPU. We numerically study the performance and accuracy of the rigid multiblob methods for suspensions in an unbounded domain in Section 4, and study particles sedimented near a no-slip boundary in Section 5. We find that resolving spherical particles with 12 blobs placed on the vertices of an icosahedron [129] is notably more accurate than the FTS (force-torque-stresslet plus degenerate quadrupole) truncation typically employed in Stokesian dynamics simulations, provided that the effective hydrodynamic radius of the rigid multiblob is adjusted to account for the finite size of the blobs. We also find that a small number of iterations of a Krylov method are required to solve the required linear system, and importantly, the number of iterations is constant *independent* of the number of rigid bodies, making it possible to develop a linear or near-linear scaling algorithm. For *resistance problems*, however, we observe a number of iterations growing at least as fast as the linear dimensions of the system. This is consistent with similar studies of iterative solvers for Stokesian dynamics by Ichiki [65].

For confined systems, however, even in the simplest case of a periodic system, the Green's function for Stokes flow and the associated RPY tensor is difficult to obtain in closed form, and when it is possible to write an analytical result, the resulting formulas are typically based on infinite series that are expensive to evaluate. For periodic systems this is commonly addressed by using Ewald summation [10] based on the fast Fourier transform (FFT) [126]; the present state of the art for Stokes flow is the spectral Ewald method [90], which has recently been used for Stokesian dynamics simulations of periodic suspensions [132]. A key deficiency of most existing methods is that they rely critically on having triply periodic domains and the use of the FFT. Generalizing these methods to nonperiodic domains while keeping their linear scaling requires a large development effort and typically a new implementation for every different geometry [57; 96]. Furthermore, in a number of applications involving active particles [94; 105], there is a surface slip (e.g., electrohydrodynamic or osmophoretic flow) induced on the bottom boundary due to the gradients created by the particles, and this slip drives or at least strongly affects the motion of the particles. Accounting for this slip requires solving an additional equation such as a Poisson or Laplace equation for the electric potential or concentration of chemical fuel with nontrivial boundary conditions on the particle and wall surfaces. The solution of this additional equation provides the slip boundary condition for the Stokes equations, which must be solved to find the resulting fluid

flow and active particle motion. Such nontrivial multiphysics coupling is quite hard to accomplish in existing methods.

To address these difficulties, in Section 3 we develop a method for general cuboidal confined domains which does not require analytical Green's functions. This relies on an immersed boundary (IB) method for obtaining an approximation to the RPY tensor in confined geometries, as recently developed by some of the authors [33]. This technique has been combined with the concept of multiblob representation of rigid bodies in a follow-up work [129], but in this work stiff elastic springs were used to enforce the rigidity. By contrast, we ensure the rigidity of the multiblobs via Lagrange multipliers which are solved concurrently with solving for the fluid pressure and velocity. Our key novel contribution is an effective preconditioner for the rigidly constrained Stokes problem in periodic and nonperiodic domains, obtained by combining our recently developed preconditioner for a rigid-body IB method [71] with a block-diagonal preconditioner for the mobility subproblem.

In the IB method developed in Section 3 and studied numerically in Section 6, analytical Green's functions are replaced by an "on-the-fly" computation carried out by a standard finite-volume fluid solver. This Stokes solver can readily handle nontrivial boundary conditions; for example, slip along the walls [94; 105] can easily be accounted for. Furthermore, suspensions at small but nonzero Reynolds numbers can be handled with little extra work [7; 71]. Additionally, we avoid uncontrolled approximations relying on truncations of multipole expansions to a fixed order [117; 92; 7; 50], and we can seamlessly handle arbitrary body shapes and deformation kinematics. Lastly, and importantly, in the spirit of fluctuating hydrodynamics [33; 73; 4], it is straightforward to generate the stochastic increments required to simulate the Brownian motion of small rigid particles suspended in a fluid by including a fluctuating stress in the fluid equations, as we will discuss in more detail in future work; here we focus on the deterministic mobility and resistance problems. At the same time, our method also has some disadvantages compared to methods such as boundary integral or boundary element methods. Notably, it requires filling the domain with a dense uniform fluid grid, which is expensive at low densities. It is also a low-order method that cannot compute solutions as accurately as spectral boundary integral formulations. We do believe, nevertheless, that the method developed here offers a good compromise between accuracy, efficiency, scalability, flexibility, and extensibility, compared to other more specialized formulations.

We apply our methods to a number of test problems for which analytical solutions are known, but also study a few nontrivial problems that have not been properly addressed in the literature. In Section 5.2 we study the mobility of a cylinder of finite aspect ratio that is parallel to a no-slip boundary and compare to experimental measurements and asymptotic theory based on a slender-body approximation. In

Section 5.3 we study the formation of a stable rotating pair of active "extensor" or "pusher" nanorods next to a no-slip boundary, and confirm the direction of rotation observed in recent experiments [29]. In Section 6.4 we compute the effective diffusion coefficient of a boomerang-shaped colloid in a slit channel, and compare to recent experimental measurements [21; 22]. In Section 6.6 we study the mean and variance of the sedimentation velocity in a binary suspension of spheres of size ratio 2, and compare to recent Stokesian dynamics simulations [131; 132].

2. Rigid multiblob models of colloidal suspensions

In this section we develop the rigid multiblob model of colloidal particles at zero Reynolds number. The kind of models we use here are not new, but we present the method in detail instead of relying on previous presentations, the most relevant of which are those of Swan et al. [125; 126]. This is in part to present the formulation in our notation, and in part to explain the differences with other closely related methods. Our key novel contribution in this section is the preconditioned iterative solver described in Section 2.2; the performance and scaling of our mobility solver is studied numerically for unbounded domains in Section 4.4, and for particles confined near a single wall in Section 5.4.

The modeling of suspensions of rigid spheres at small Reynolds numbers is a well-developed field with a long history. A powerful class of methods are related to Brownian dynamics with hydrodynamic interactions (BDHI) [68; 58; 108; 75] and Stokesian dynamics (SD) [117; 122; 62; 123; 8; 132] (note that these terms are used differently in different communities). The difference between these two (as we define them here) is that BDHI uses what we call a minimally resolved model [33] in which each colloid (for colloidal suspensions) or polymer bead (for polymeric suspensions) is only resolved at the monopole level, more precisely, at the Rotne–Prager level [126]. By contrast, in SD the next level in a multipole expansion is taken into account and torques and stresslets are also accounted for. It has been shown recently that yet one more order needs to be kept in the multipole expansion to model suspensions of active spheres [50; 119], and a suitable Galerkin truncation of the multipole hierarchy has been developed for active spheres in unbounded domains [119], as well as for active spheres confined near a no-slip boundary [118]. It is also possible to account for higher-order multipoles [24; 26; 119; 81; 82], leading to more complicated (and computationally expensive) but also more accurate models. It has also been shown that multipole expansions converge very poorly for nearly touching spheres due to the divergence of the lubrication forces, and in most methods for dense colloidal suspensions of hard spheres pairwise lubrication corrections are added in a somewhat ad hoc manner; we will refer to this approach as SD with lubrication.

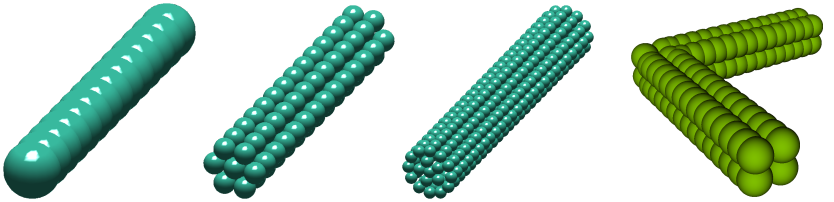


Figure 1. Rigid multiblob models of colloidal particles manufactured in recent experimental work. Left three panels: a cylinder of aspect ratio of about 6, similar to the active nanorods studied experimentally in [127; 29], for three different resolutions: minimally resolved model with 14 blobs, marginally resolved model with 86 blobs, and well-resolved model with 324 blobs. Right: a 120-blob model of a boomerang with square cross-section, as studied experimentally in [21].

Given the well-developed tools for modeling sphere suspensions, it is natural to leverage them when modeling suspensions of particles of more complex shapes. Here we describe a technique capable of, in principle, modeling passive rigid particles of arbitrary shape. The method can also be used to model, without any extra effort, active particles with active slip layers, i.e., particles which are phoretic (e.g., osmophoretic, electrophoretic, chemophoretic, etc.) due to an apparent slip at their surface. For the purposes of hydrodynamic calculations, we discretize rigid bodies by constructing them out of multiple rigidly connected spherical "blobs" or beads of hydrodynamic radius a. These blobs can be thought of as hydrodynamically minimally resolved spheres forming a rigid conglomerate that approximates the hydrodynamics of the actual rigid object being studied. We prefer the word "blob" over "sphere" or "point" or "monopole" because blobs are not spheres as they do not have a well-defined surface like spheres do, they have a finite size associated with them (the hydrodynamic blob radius a) unlike points, and they account for a degenerate quadrupole associated to the Faxén corrections in addition to a force monopole. The word "bead" is also appropriate, but we prefer to reserve that for polymer models (bead-spring or bead-link models).

Examples of "multiblob" [129] models of two types of colloidal particles are illustrated in Figure 1. In the leftmost panel, we show a minimally resolved model of a rigid rod, with dimensions similar to active metallic "nanorods" used in recent experiments [127; 29]. In this minimally resolved model the blobs, shown as spheres with radius equal to a, are placed in a row along the axes of the cylinder. Such minimally resolved models are particularly suited for cylinders of large but finite aspect ratio; for very thin rods such as actin filaments boundary integral methods based on slender-body theory [102] will be more effective. In the more resolved model illustrated in the second panel from the left, a hexagon of blobs is placed around the circumference of the cylinder to better resolve it. A yet more resolved model with a dodecagon of blobs around the cylinder circumference is shown in

the third panel from the left. In the rightmost panel of Figure 1 we show a blob model of a colloidal boomerang with a square cross-section, as manufactured using lithography and studied in [21]. Similar "bead" or "raspberry" models appear in a number of studies of hydrodynamics of particle suspensions [104; 48; 46; 110; 91; 100; 125; 62; 25; 80; 18; 129; 106; 136].

In many studies, stiff elastic springs between the blobs are used to keep the structure rigid; in some models the fluid or particle inertia is included also. Here, we keep the structures *strictly rigid* and refer to the resulting structures as *rigid multiblob* models. Such rigid multiblob models have been used in a number of prior studies [104; 48; 46; 125; 62; 25; 80; 28], but we refer to [125] for a detailed exposition. Our primary focus in this section will be to develop algorithmic techniques that allow suspensions of tens or even hundreds of thousands of rigid multiblob particles to be simulated efficiently. This is in many ways primarily an exercise in numerical linear algebra, but one that is *necessary* to make the rigid multiblob approach useful for simulating moderately dense suspensions. A second goal, which will be realized in the results sections of this paper, will be to carefully assess the accuracy of rigid multiblob models as a function of their resolution (number of blobs per body).

2.1. ***Hydrodynamics of rigid multiblobs.*** We now summarize the main equations used to solve the mobility and resistance problems for a collection of rigid multiblobs immersed in a viscous fluid. We first discuss the hydrodynamic interaction between blobs, and then discuss the hydrodynamic interactions between rigid bodies.

In the notation used below, we will use the Latin indices i, j, k, and l for individual blobs, and reserve Latin indices p, q, r, and s for bodies. We will denote by $\mathcal{B}_p$ the set of blobs comprising body p. We will consider a suspension of N rigid bodies with a chosen reference *tracking point* on body p having position $\boldsymbol{q}_p$, and the orientation of body p relative to a *reference configuration* represented by the quaternion $\boldsymbol{\theta}_p$ [34]. The linear velocity of (the chosen tracking point on) body p will be denoted by $\boldsymbol{u}_p$, and its angular velocity will be denoted by $\boldsymbol{\omega}_p$. The total force applied on body p is $\boldsymbol{f}_p$, and the total torque is $\boldsymbol{\tau}_p$. The composite configuration vector of position and orientation of body p will be denoted by $\boldsymbol{Q}_p = \{\boldsymbol{q}_p, \boldsymbol{\theta}_p\}$, the composite vector of linear and angular velocity by $\boldsymbol{U}_p = \{\boldsymbol{u}_p, \boldsymbol{\omega}_p\}$, and the composite vector of forces and torques by $\boldsymbol{F}_p = \{\boldsymbol{f}_p, \boldsymbol{\tau}_p\}$. The position of blob $i \in \mathcal{B}_p$ will be denoted by $\boldsymbol{r}_i$, and its velocity will be denoted by $\dot{\boldsymbol{r}}_i$. When not subscripted, vectors will refer to the composite vector formed by all bodies or all blobs on all bodies. For example, $\boldsymbol{U}$ will denote the linear and angular velocities of all bodies, and $\boldsymbol{r}$ will denote the positions of all of the blobs. We will use a superscript to denote portions of composite vectors for all blobs belonging to one body; for example, $\boldsymbol{r}^{(p)} = \{\boldsymbol{r}_i \mid i \in \mathcal{B}_p\}$ will denote the vector of positions of all blobs belonging to body p.

The fact that the multiblob p is rigid is expressed by the "no-slip" kinematic condition

$$\dot{\boldsymbol{r}}_i = \boldsymbol{u}_p + \boldsymbol{\omega}_p \times (\boldsymbol{r}_i - \boldsymbol{q}_p) \quad \text{for all } i \in \mathcal{B}_p. \tag{2}$$

This no-slip condition can be written for all bodies succinctly as

$$\dot{\boldsymbol{r}} = \mathcal{K}\boldsymbol{U}, \tag{3}$$

where $\mathcal{K}(\boldsymbol{Q})$ is a simple geometric matrix [126]. We will denote the apparent velocity of the fluid at point $\boldsymbol{r}_i$ by $\boldsymbol{w}_i \approx \boldsymbol{v}(\boldsymbol{r}_i)$. For a *passive blob*, i.e., a blob that represents a passive part of the rigid particle, the *no-slip* boundary condition requires that $\boldsymbol{w}_i = \dot{\boldsymbol{r}}_i$. However, for *active blobs* an additional apparent slip of the fluid relative to the surface of the body can be imposed, resulting in a nonzero *slip* $\breve{\boldsymbol{u}}_i = \boldsymbol{w}_i - \dot{\boldsymbol{r}}_i$. This kind of active propulsion is termed "implicit swimming gait" by Swan and Brady [122]. An "explicit swimming gait" [122] can be taken into account without any modifications to the formulation or algorithm by simply replacing (2) with

$$\boldsymbol{w}_i = \dot{\boldsymbol{r}}_i = \boldsymbol{u}_p + \boldsymbol{\omega}_p \times (\boldsymbol{r}_i - \boldsymbol{q}_p) + \breve{\boldsymbol{u}}_i. \tag{4}$$

That is, the only difference between "slip" and "deformation" is whether the blobs move relative to the rigid body frame dragging the fluid along, or stay fixed in the body frame while the fluid passes by them. One can of course even combine the two and have the blobs move relative to the rigid body while also pushing flow; for example, this can be used to model an active filament where there is slip along the filament but the filament itself is moving. In the end, the only thing that matters to the formulation is the velocity difference

$$\breve{\boldsymbol{u}}_i \approx \boldsymbol{v}(\boldsymbol{r}_i) - (\boldsymbol{u}_p + \boldsymbol{\omega}_p \times (\boldsymbol{r}_i - \boldsymbol{q}_p)). \tag{5}$$

In Appendix C we explain how to model permeable (porous) bodies by making the apparent slip proportional to the fluid-blob force $\boldsymbol{\lambda}$.

The fundamental problem tackled in this paper is the solution of the *mobility problem*, that is, the computation of the motion of the bodies given the applied forces and torques on the bodies and the slip velocity. Because of the linearity of the Stokes equations and the boundary conditions, there exists an affine linear mapping

$$\boldsymbol{U} = \mathcal{N}\boldsymbol{F} - \breve{\mathcal{M}}\breve{\boldsymbol{u}},$$

where the *body mobility matrix* $\mathcal{N}(\boldsymbol{Q})$ depends on the configuration and is the central object of the computation. The *active mobility matrix* $\breve{\mathcal{M}}$ is a discretization of the active mobility operator $\breve{\mathcal{N}}$, and gives the active motion of force- and torque-free particles. Note that $\breve{\mathcal{M}}$ is related to, but different from, the propulsion matrix introduced in [119]. The propulsion matrix is essentially a finite-dimensional

projection of the operator $\breve{\mathcal{N}}$ that only depends on the choice of basis functions used to express the surface slip velocity $\breve{\boldsymbol{u}}$, and does not depend on the specific discretization of the body or quadrature rules, as does $\breve{\mathcal{M}}$.

In the remainder of this section we develop a method for computing $\boldsymbol{U}$ given $\boldsymbol{F}$ and $\breve{\boldsymbol{u}}$, i.e., a method for computing the combined action of $\mathcal{N}$ and $\breve{\mathcal{M}}$, for large collections of nonoverlapping rigid particles. We will also briefly discuss the *resistance problem*, in which we are given the motion of the bodies as a specified kinematics, and seek the resulting drag forces and torques, which have the form

$$\boldsymbol{F} = \mathcal{R}\boldsymbol{U} + \breve{\mathcal{R}}\,\breve{\boldsymbol{u}},$$

where the *body resistance matrix* $\mathcal{R} = \mathcal{N}^{-1}$ and $\breve{\mathcal{R}} = \mathcal{N}^{-1}\breve{\mathcal{M}}$ is the *active resistance matrix*.

Blob mobility matrix. The blob-blob translational mobility matrix $\mathcal{M}$ describes the hydrodynamic interactions between the N_b blobs, accounting for the influence of the boundaries. Specifically, if the blobs are free to move (i.e., not constrained rigidly) with the fluid under the action of the set of translational forces $\boldsymbol{\lambda}_i$, the translational velocities of the blobs will be

$$\boldsymbol{w} = \dot{\boldsymbol{r}} + \breve{\boldsymbol{u}} = \mathcal{M}\boldsymbol{\lambda}. \tag{6}$$

The mobility matrix $\mathcal{M}$ is a block matrix of dimension $(dN_b) \times (dN_b)$, where d is the dimensionality. The $d \times d$ block $\mathcal{M}_{ij}$ computes the velocity of blob i given the force on blob j, neglecting the presence of the other blobs in a *pairwise* approximation.

To construct a suitable $\mathcal{M}$, we can think of blobs as spheres of hydrodynamic radius a. For two well-separated spheres i and j of radius a we have the far-field approximation [75; 122; 130]

$$\mathcal{M}_{ij} \approx \eta^{-1}(\boldsymbol{I} + \tfrac{1}{6}a^2\nabla^2_{\boldsymbol{r}'})(\boldsymbol{I} + \tfrac{1}{6}a^2\nabla^2_{\boldsymbol{r}''})\mathbb{G}(\boldsymbol{r}', \boldsymbol{r}'')\big|^{\boldsymbol{r}'=\boldsymbol{r}_j}_{\boldsymbol{r}''=\boldsymbol{r}_i}, \tag{7}$$

where η is the fluid viscosity and $\mathbb{G}$ is the Green's function for the steady Stokes problem with unit viscosity, with the appropriate boundary conditions such as no-slip on the boundaries of the domain. The differential operator $\boldsymbol{I} + (a^2/6)\nabla^2$ is called the Faxén operator [75]. Note that the form of (7) guarantees that the mobility matrix is symmetric-positive-semidefinite (SPD) by construction since $\mathbb{G}$ is an SPD kernel.

For a three-dimensional unbounded domain with fluid at rest at infinity, the Green's function is isotropic and given by the Oseen tensor:

$$\mathbb{G}(\boldsymbol{r}', \boldsymbol{r}'') \equiv \mathbb{O}(\boldsymbol{r} = \boldsymbol{r}' - \boldsymbol{r}'') = \frac{1}{8\pi r}\Big(\boldsymbol{I} + \frac{\boldsymbol{r}\otimes\boldsymbol{r}}{r^2}\Big). \tag{8}$$

Using this expression in (7) yields the far-field component of the Rotne–Prager–Yamakawa (RPY) tensor [113], commonly used in BDHI. A correction needs to be introduced when particles are close to each other to ensure an SPD mobility matrix [113], which can be derived by using an integral form of the RPY tensor valid even for overlapping particles [130], to give

$$\mathcal{M}_{ij} = \frac{1}{6\pi\eta a}\begin{cases} C_1(r_{ij})\boldsymbol{I} + C_2(r_{ij})(\boldsymbol{r}_{ij}\otimes\boldsymbol{r}_{ij})/r_{ij}^2, & r_{ij} > 2a, \\ C_3(r_{ij})\boldsymbol{I} + C_4(r_{ij})(\boldsymbol{r}_{ij}\otimes\boldsymbol{r}_{ij})/r_{ij}^2, & r_{ij} \le 2a, \end{cases} \tag{9}$$

where $\boldsymbol{r}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j$ and

$$C_1(r) = \frac{3a}{4r} + \frac{a^3}{2r^3}, \qquad C_2(r) = \frac{3a}{4r} - \frac{3a^3}{2r^3}, \qquad C_3(r) = 1 - \frac{9r}{32a}, \qquad C_4(r) = \frac{3r}{32a}.$$

The diagonal blocks of the mobility matrix, i.e., the self-mobility, can be obtained by setting $r_{ij} = 0$ to obtain $\mathcal{M}_{ii} = (6\pi\eta a)^{-1}\boldsymbol{I}$, which matches the Stokes solution for the drag on a translating sphere; this is an important continuity property of the RPY tensor [39]. We will use the RPY tensor (9) for simulations of rigid-particle suspensions in unbounded domains in Section 4.

In principle, it is possible to generalize the RPY tensor to any flow geometry, i.e., to any boundary conditions (and imposed external flow) [130], including periodic domains [10; 99], as well as confined domains [122; 123]. However, we are not aware of any tractable analytical expressions for the complete RPY tensor (including near-field corrections) even for the simplest confined geometry of particles near a single no-slip boundary. In the presence of a single no-slip wall, an analytic approximation to $\mathcal{M}_{ij}$ is given by Swan and Brady [122] (and rederived later in [49]) as a generalization of the Rotne–Prager (RP) tensor [113] to account for the no-slip boundary using Blake's image construction [13]. As shown in [130], the corrections to the Rotne–Prager tensor (7) for particles that overlap each other but not the wall are independent of the boundary conditions, and are thus given by the standard RPY expressions (9) for unbounded domains. Therefore, in Section 5 we compute $\mathcal{M}$ by adding to the RPY tensor (9) wall corrections corresponding to the translation-translation part of the Rotne–Prager–Blake mobility given by (B1) and (C2) in [122], ignoring the higher-order torque and stresslet terms in the spirit of the minimally resolved blob model. The expressions derived by Swan and Brady [122] assume that neither particle overlaps the wall and the resulting expressions are not guaranteed to lead to an SPD $\mathcal{M}$ if one or more blobs overlap the wall, as we discuss in more detail in the conclusions.

For more complicated geometries, such as a slit or a square (duct) channel, analytical computations of the Green's function become quite complicated and tedious, and numerical computations typically require pretabulations [75; 123; 12]. In Section 6 we explain how a grid-based finite-volume Stokes solver can be used

to obtain the action of the Green's function and thus compute the action of the mobility matrix for confined domains, for essentially arbitrary combinations of periodic, free-slip, no-slip, or stress boundary conditions.

Body mobility matrix. After discretizing the rigid bodies as rigid multiblobs, we can write down a system of equations that constrain the blobs to move rigidly in a straightforward manner. Letting $\boldsymbol{\lambda}$ be a vector of forces (Lagrange multipliers) that acts on each blob to enforce the rigidity of the body, we have the following linear system for $\boldsymbol{\lambda}$, $\boldsymbol{u}$, and $\boldsymbol{\omega}$ for all bodies p:

$$\begin{aligned} \sum_j \mathcal{M}_{ij}\boldsymbol{\lambda}_j &= \boldsymbol{u}_p + \boldsymbol{\omega}_p \times (\boldsymbol{r}_i - \boldsymbol{q}_p) + \breve{\boldsymbol{u}}_i \quad \text{for all } i \in \mathcal{B}_p, \\ \sum_{i\in\mathcal{B}_p} \boldsymbol{\lambda}_i &= \boldsymbol{f}_p, \\ \sum_{i\in\mathcal{B}_p} (\boldsymbol{r}_i - \boldsymbol{q}_p) \times \boldsymbol{\lambda}_i &= \boldsymbol{\tau}_p. \end{aligned} \tag{10}$$

The first equation is the no-slip condition obtained by combining (6) and (2). The second and third equations are the force and torque balance conditions for body p. Note that the physical interpretation of $\boldsymbol{\lambda}$ is a total force on the portion of the surface of the body associated with a given blob. If one wants to think of (10) as a regularized discretization of the first-kind integral equation (A-5) and obtain a pointwise value of the traction force *density*, one should divide $\boldsymbol{\lambda}_j$ by the surface area ΔA_j associated with blob j, which plays the role of a quadrature weight [28]; we will discuss more sophisticated quadrature rules [120; 101] in the conclusions.

We can write the *mobility problem* (10) in compact matrix notation as a *saddle-point* linear system of equations for the rigidity forces $\boldsymbol{\lambda}$ and unknown motion $\boldsymbol{U}$:

$$\begin{bmatrix} \mathcal{M} & -\mathcal{K} \\ -\mathcal{K}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{U} \end{bmatrix} = \begin{bmatrix} \breve{\boldsymbol{u}} \\ -\boldsymbol{F} \end{bmatrix}. \tag{11}$$

Forming the Schur complement by eliminating $\boldsymbol{\lambda}$ we get (see also (1) in [125] or (32) in [126])

$$\boldsymbol{U} = \mathcal{N}\boldsymbol{F} - (\mathcal{N}\mathcal{K}^T\mathcal{M}^{-1})\breve{\boldsymbol{u}} = \mathcal{N}\boldsymbol{F} - \breve{\mathcal{M}}\,\breve{\boldsymbol{u}},$$

where the body mobility matrix $\mathcal{N}$ is

$$\mathcal{N} = (\mathcal{K}^T\mathcal{M}^{-1}\mathcal{K})^{-1}, \tag{12}$$

and is evidently SPD since $\mathcal{M}$ is. Although written in this form using the inverse of $\mathcal{M}$, unlike in a number of prior works [104; 46; 62; 25; 80], we obtain $\boldsymbol{U}$ by solving (11) directly using an iterative solver, as we explain in more detail in Section 2.2. We note that one can compute a fluid velocity field $\boldsymbol{v}(\boldsymbol{r})$ from $\boldsymbol{\lambda}$ using a procedure we describe in Appendix B.

The *resistance problem*, on the other hand, consists of solving for $\boldsymbol{\lambda}$ in

$$\mathcal{M}\boldsymbol{\lambda} = \mathcal{K}\boldsymbol{U} + \breve{\boldsymbol{u}}, \tag{13}$$

and then computing $\boldsymbol{F} = \mathcal{K}^T\boldsymbol{\lambda}$, giving

$$\boldsymbol{F} = (\mathcal{K}^T\mathcal{M}^{-1}\mathcal{K})\boldsymbol{U} + (\mathcal{K}^T\mathcal{M}^{-1})\breve{\boldsymbol{u}} = \mathcal{R}\boldsymbol{U} + \breve{\mathcal{R}}\,\breve{\boldsymbol{u}}.$$

At first glance, it appears that solving the resistance system (13) is easier than solving the saddle-point problem (11); however, as we explain in more detail in Section 4.4, the mobility problem is significantly easier to solve using iterative methods than the resistance problem, consistent with similar observations in the context of Stokesian dynamics [65]. Observe that the saddle-point formulation (11) applies more broadly to *mixed* mobility/resistance problems, where some of the rigid-body degrees of freedom are constrained but some are free [30]. An example is a suspension of spheres being rotated by a magnetic field at a specified angular velocity but free to move translationally, or a suspension of colloids fixed in space by strong laser tweezers but otherwise free to rotate, or even a hinged body that can only move in a partially constrained manner. In cases such as these we simply redefine $\boldsymbol{U}$ to contain the free kinematic degrees of freedom and modify the definition of the kinematic matrix $\mathcal{K}$. Much of what we say below continues to apply, but with the caveat that the expected speed of convergence of iterative methods is expected to depend on the nature of the imposed constraints, as we discuss in Section 4.4.

Note that (12) is somewhat formal, and in practice all inverses should be replaced by pseudoinverses. For instance, in the limit when infinitely many blobs cover the surface of a body, the mobility matrix $\mathcal{M}$ is not invertible since making $\boldsymbol{\lambda}$ perpendicular to the surface will not yield any flow because it will try to compress the (fictitious) incompressible fluid inside the body. Note that this nontrivial null space of the mobility poses no problem when using an iterative method to solve (11) because the right-hand side is in the proper range due to the imposition of the volume-preservation constraint (A-6). It is also possible that the matrix $\mathcal{K}^T\mathcal{M}^{-1}\mathcal{K}$ is not invertible. A typical example for this is the minimally resolved cylinder shown in the leftmost panel of Figure 1. Because all of the forces $\boldsymbol{\lambda}$ are applied exactly on the semiaxes of the cylinder, they cannot exert a torque around the symmetry axes of the rod. Again, there is no problem with iterative solvers for (11) if the applied force is in the appropriate range (e.g., one should not apply a torque around the semiaxes of a minimally resolved cylinder).

2.2. ***Iterative mobility solver.*** For a small number of blobs, (11) can be solved by direct inversion of $\mathcal{M}$, as done in most prior works. For large systems, which are the focus of our work, iterative methods are required. A standard approach used in

the literature is to eliminate one of the variables $\boldsymbol{\lambda}$ or $\boldsymbol{U}$. Eliminating $\boldsymbol{\lambda}$ leads to the equation

$$(\mathcal{K}^T\mathcal{M}^{-1}\mathcal{K})\boldsymbol{U} = \boldsymbol{F} - \mathcal{K}^T\mathcal{M}^{-1}\breve{\boldsymbol{u}}, \tag{14}$$

which requires the action of $\mathcal{M}^{-1}$, which must itself be obtained inside a nested iterative solver, increasing both the complexity and the cost of the method. Swan and Wang [126] have recently used the conjugate gradient method to solve (14), preconditioning using the block-diagonal matrix $\mathcal{P} = (6\pi\eta a)(\mathcal{K}^T\mathcal{K})$.

An alternative is to write a system equivalent to (11), for an arbitrary constant $c \neq 0$,

$$\begin{bmatrix} \mathcal{M} & -\mathcal{K} \\ -\mathcal{K}^T(\mathcal{I} + c\mathcal{M}) & c(\mathcal{K}^T\mathcal{K}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{U} \end{bmatrix} = \begin{bmatrix} \breve{\boldsymbol{u}} \\ -(\boldsymbol{F} + c\mathcal{K}^T\breve{\boldsymbol{u}}) \end{bmatrix}, \tag{15}$$

from which we can easily eliminate $\boldsymbol{U}$ to obtain an equation for $\boldsymbol{\lambda}$ only, in the form

$$[\mathcal{M}(\mathcal{I} - \mathcal{K}(\mathcal{K}^T\mathcal{K})^{-1}\mathcal{K}^T) - c^{-1}\mathcal{K}(\mathcal{K}^T\mathcal{K})^{-1}\mathcal{K}^T]\boldsymbol{\lambda} = \text{rhs}, \tag{16}$$

where we omit the full expression for the right-hand side for brevity. The system (16) can now be solved using (preconditioned) conjugate gradients, and only requires the inverse of the simpler matrix $\mathcal{K}^T\mathcal{K}$. Note that, although not presented in this way, this is the essence of the approach that is followed and recommended by Swan et al. [125] (see the appendices of [125] and note that c is denoted by λ in that paper); they recommend computing the action of $(\mathcal{K}^T\mathcal{K})^{-1}$ with an iterative method preconditioned by an incomplete Cholesky factorization. A similar approach is followed in boundary integral formulations (which are usually formulated using a double-layer density), where a continuum operator related to $\mathcal{K}(\mathcal{K}^T\mathcal{K})^{-1}\mathcal{K}^T$ is computed and then discretized using a quadrature rule [111; 77].

In contrast to the approaches taken by Swan et al. [125; 126], we have found that numerically the best approach to solving for the unknown rigid-body motions of the particles is to solve the extended saddle-point problem (11) for *both* $\boldsymbol{U}$ and $\boldsymbol{\lambda}$ *directly*, using a preconditioned iterative Krylov method. In fact, as we will demonstrate in the results section of this paper, such an approach has computational complexity that is essentially linear in the number of blobs because the number of iterations required to solve (11) is quite modest when an appropriate preconditioner, described below, is used. This approach does not require computing (the action of) $(\mathcal{K}^T\mathcal{K})^{-1}$ and leads to a very simple implementation.

Matrix-vector product. A Krylov solver for (11) requires two components:

- an efficient algorithm for performing the matrix-vector product, which in our case amounts to a fast method to multiply the dense but low-rank mobility matrix $\mathcal{M}$ by a vector of blob forces $\boldsymbol{\lambda}$, and
- a suitable preconditioner, which is an approximate solver for (11).

How to efficiently compute $\mathcal{M}\boldsymbol{\lambda}$ depends very much on the boundary conditions and thus the form of the Green's function used to construct $\mathcal{M}$. For unbounded domains, in this work we use the fast multipole method (FMM) developed specifically for the RPY tensor in [89]; alternative kernel-independent FMMs could also be used, and have also been generalized to periodic domains [87]. The FMM method has an essentially linear computational cost of $O(N_b \log N_b)$ for a single matrix-vector multiplication. In the simulations presented here we use a fixed and rather tight relative tolerance for the FMM $\sim 10^{-9}$ throughout the iterative solution process. Krylov methods, however, allow one to *lower* the accuracy of the matrix-vector product as the residual is reduced [14]; this has recently been used to lower the cost of FMM-based boundary integral methods [133]. We will explore such optimizations in future work.

For rigid particles sedimented near a single no-slip wall, we have implemented a GPU-based direct-summation matrix-vector product based on the Rotne–Prager–Blake tensor derived by Swan and Brady [122]. This has, asymptotically, a quadratic computational cost of $O(N_b^2)$; however, the computation is trivially parallel so the multiplication is remarkably fast even for one million blobs because of the very large number of threads available on modern GPUs. Gimbutas et al. have recently developed an FMM method for the Blake tensor by using a simple image construction (image Stokeslet plus a harmonic scalar correction) and applying an infinite-space FMM method to the extended system of singularities [51]. However, this construction has not yet been generalized to the Rotne–Prager–Blake tensor, and furthermore, the FMM will not be more efficient than the direct product on GPUs in practice unless a large number of blobs is considered. For fully confined domains, we will adopt an extended saddle-point formulation that will be described in Section 6.

Preconditioner. In this work we demonstrate that a very efficient yet simple preconditioner for (11) is obtained by neglecting hydrodynamic interactions between different bodies, that is, setting the elements of $\mathcal{M}$ corresponding to pairs of blobs on *distinct* bodies to zero in the preconditioner. This amounts to making a block-diagonal approximation of the mobility $\widetilde{\mathcal{M}}$ defined by only keeping the diagonal blocks corresponding to a single body interacting *only* with the boundaries of the domain:

$$\widetilde{\mathcal{M}}^{(pq)} = \delta_{pq}\mathcal{M}^{(pp)}. \tag{17}$$

We will demonstrate here that the *indefinite block-diagonal* preconditioner,

$$\mathcal{P} = \begin{bmatrix} \widetilde{\mathcal{M}} & -\mathcal{K} \\ -\mathcal{K}^T & \mathbf{0} \end{bmatrix}, \tag{18}$$

is a very effective preconditioner for solving (11).

Applying the preconditioner (18) amounts to solving the linear system

$$\begin{bmatrix} \widetilde{\mathcal{M}} & -\mathcal{K} \\ -\mathcal{K}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{U} \end{bmatrix} = \begin{bmatrix} \breve{\boldsymbol{u}} \\ -\boldsymbol{F} \end{bmatrix}, \tag{19}$$

which is quite easy to do since the approximate body mobility matrix (Schur complement),

$$\widetilde{\mathcal{N}} = (\mathcal{K}^T \widetilde{\mathcal{M}}^{-1} \mathcal{K})^{-1},$$

is itself a block-diagonal matrix where each block on the diagonal refers to a single body neglecting all hydrodynamic interactions with other bodies:

$$\widetilde{\mathcal{N}}_{pq} = \delta_{pq} ((\mathcal{K}^{(p)})^T (\mathcal{M}^{(pp)})^{-1} \mathcal{K}^{(p)})^{-1}.$$

Computing $\widetilde{\mathcal{N}}_{pq}$ requires a dense matrix inversion (e.g., Cholesky factorization) of the much smaller mobility matrix $\mathcal{M}^{(pp)}$, whose size is $(dN_b^{(p)}) \times (dN_b^{(p)})$, where $N_b^{(p)}$ is the number of blobs on body p. In the case of an infinite domain, the factorization of $\mathcal{M}^{(pp)}$ can be precomputed once at the beginning of a dynamic simulation and reused during the simulation due to the rotational and translational invariance of the RPY tensor; one only needs to apply rotation matrices to the right-hand side and the result to convert between the original reference configuration of the body and the current configuration. Furthermore, particles of the same shape and size discretized with the same number of blobs as body p can share a single factorization of $\mathcal{M}^{(pp)}$ and $\widetilde{\mathcal{N}}_{pp}$. In cases where $\mathcal{M}^{(pp)}$ depends in a nontrivial way on the position of the body, as for (partially) confined domains, one needs to factorize $\mathcal{M}^{(pp)}$ for all bodies p at every time step; this factorization can still be reused during the iterative solve in each application of the preconditioner.

Because our preconditioner is indefinite, one cannot use the preconditioned conjugate gradient (PCG) Krylov method to solve (11) without modification. One of the most robust iterative methods, which we use in this work, is the generalized minimum residual method (GMRES). The key advantage of GMRES is that it is guaranteed to reduce the residual from iteration to iteration. Its main downside is that it requires storing a large number of intermediate vectors (i.e., the history of the iterates). GMRES also can stall, although this can be corrected to some extent by restarts. An alternative to GMRES is the (stabilized) biconjugate gradient (BiCG(Stab)) method, which works for nonsymmetric matrices as well. In our implementation we have relied on the PETSc library [5] for iterative solvers; this library makes it very easy to experiment with different iterative solvers.

3. Rigid multiblobs in confined domains

The rigid multiblob method described in Section 2 requires a technique for multiplying the blob-blob mobility matrix with a vector. Therefore, this approach,

like all other Green's-function-based methods [24; 81; 82; 26; 123; 122; 119; 77; 96; 90; 88; 2; 75; 57], is very geometry-specific and does not generalize easily to more complicated boundary conditions. To handle geometries for which there is no simple analytical expression for the Green's function, such as slit or square channels, pretabulation of the Green's function is necessary, and ensuring a positive semidefinite mobility matrix is in general difficult. Another difficulty with Green's-function-based methods is that including a "background" flow is only simple when this flow can be computed easily analytically, such as simple shear flows. But for more complicated geometries, such as Poiseuille flow through a square channel, computing the base flow is itself not trivial or requires evaluating expensive infinite-series solutions.

An alternative approach is to use a traditional Stokes solver to solve the fluid equations numerically [58]. This requires filling the domain with a grid, which can increase the number of degrees of freedom considerably over just discretizing the surface of the immersed bodies. However, the number of fluid degrees of freedom can be held approximately constant as more bodies are included, so that the methods typically scale very well with the number of particles and are well-suited to dense particle suspensions. Previous work [33; 4; 3] has shown how to use an immersed boundary (IB) method [107] to obtain the action of the Green's function in complex geometries. In this approach, spherical particles are minimally resolved using only a single blob per particle. In subsequent work this approach was extended to multiblob models [129], but the rigidity constraint was imposed only approximately using stiff springs, leading to numerical stiffness. A class of related minimally resolved methods based on the force coupling method (FCM) [97; 92; 73; 31] can include also torques and stresslets, as well as particle activity [32], but a number of these methods have relied strongly on periodic boundaries since they use the fast Fourier transform (FFT) to solve the (fluctuating) Stokes equations.

In recent work [71], some of the authors have developed an IB method for rigid bodies. This method applies to a broad range of Reynolds numbers. In the case of zero Reynolds number it becomes equivalent to the rigid multiblob method presented in Section 2, but with a blob-blob mobility that is computed by the fluid solver. In [71] only rigid bodies with specified motion (kinematics) were considered; here we extend the method to handle freely moving rigid bodies in Stokes flow. We will present here the key ideas and focus on the new components necessary to solve for the unknown motion of the particles; we refer the reader interested in more technical details to [33; 71]. The key novel contribution of our work is the preconditioner described in Section 3.3; the performance and scalability of our preconditioned iterative solvers are studied numerically in Section 6.5. To begin, we present a semicontinuum formulation where the relation to Section 2 is most obvious, and then we discuss the fully discrete formulation used in the actual

implementation. In Appendix C we demonstrate how to handle permeable bodies using a small modification of the formulation. Numerical results obtained using the method described here are given in Section 6.

3.1. ***Semicontinuum formulation.*** We consider here a semidiscrete model in which the rigid body has already been discretized using blobs but a continuum description is used for the fluid; that is, we consider a rigid multiblob model immersed in a continuum Stokesian fluid. In the IB literature blobs are referred to as markers, and are often thought of as "points" or "discrete delta functions". We use the term "blob", however, to connect to Section 2 and to emphasize that the blobs have a finite physical and hydrodynamic extent.

In the IB method [107] (and also the force coupling method [97]), the shape of the blob and its effective interaction with the fluid is captured through a smooth kernel function $\delta_a(\boldsymbol{r})$ that integrates to unity and whose support is localized in a region of size comparable to the blob radius a. In our rigid multiblob IB method, to obtain the fluid-blob interaction forces $\boldsymbol{\lambda}(t)$ that constrain the unknown rigid motion of the N_b blobs, we need to solve a constrained Stokes problem [71] for the fluid velocity field $\boldsymbol{v}(\boldsymbol{r}, t)$, the fluid pressure field $\pi(\boldsymbol{r}, t)$, the blob constraint forces $\boldsymbol{\lambda}(t)$, and the unknown rigid-body motions $\boldsymbol{u}(t)$ and $\boldsymbol{\omega}(t)$:

$$\begin{aligned}
\nabla\pi &= \eta\nabla^2\boldsymbol{v} + \sum_{i=1}^{N_b} \boldsymbol{\lambda}_i \delta_a(\boldsymbol{r}_i - \boldsymbol{r}), \\
\nabla\cdot\boldsymbol{v} &= 0, \\
\int \delta_a(\boldsymbol{r}_i - \boldsymbol{r}')\boldsymbol{v}(\boldsymbol{r}', t)\, d\boldsymbol{r}' &= \boldsymbol{u}_p + \boldsymbol{\omega}_p \times (\boldsymbol{r}_i - \boldsymbol{q}_p) + \breve{\boldsymbol{u}}_i \quad \text{for all } i \in \mathcal{B}_p, \\
\sum_{i\in\mathcal{B}_p} \boldsymbol{\lambda}_i &= \boldsymbol{f}_p \quad \text{for all } p, \\
\sum_{i\in\mathcal{B}_p} (\boldsymbol{r}_i - \boldsymbol{q}_p) \times \boldsymbol{\lambda}_i &= \boldsymbol{\tau}_p \quad \text{for all } p.
\end{aligned} \tag{20}$$

Note that here the velocity and pressure fields contain both the "background" and the "perturbational" contributions to the flow. In the first equation in (20), the kernel function is used to transfer (spread) the force exerted on the blob to the fluid, and in the third equation the same kernel is used to average the fluid velocity in the region covered by the blob and constrain it to follow the imposed rigid-body motion plus additional slip or body deformation. The handling of the spreading of constraint forces and averaging of the fluid velocity near physical boundaries is discussed in Appendix D of [71]. We have implicitly assumed that appropriate boundary conditions are specified for the fluid velocity and pressure. Notably, we will apply the above formulation to cases where periodic or no-slip boundary

conditions are applied along the boundaries of a cubic prism (recall that periodic boundaries are not actual physical boundaries). This includes, for example, a slit channel, a square channel, or a cubical container. It is also relatively straightforward to handle stress-based boundary conditions such as free-slip or pressure valves [53].

It is not difficult to show that (20) is equivalent to the system (10) with the mobility matrix between two blobs i and j identified with [3; 33; 71; 73; 92; 97]

$$\boldsymbol{\mathcal{M}}_{ij}(\boldsymbol{r}_i, \boldsymbol{r}_j) = \eta^{-1} \int \delta_a(\boldsymbol{r}_i - \boldsymbol{r}')\mathbb{G}(\boldsymbol{r}', \boldsymbol{r}'')\delta_a(\boldsymbol{r}_j - \boldsymbol{r}'')\, d\boldsymbol{r}'\, d\boldsymbol{r}'', \tag{21}$$

where we recall that $\mathbb{G}$ is the Green's function for the Stokes problem with unit viscosity and the specified boundary conditions. This expression can directly be compared to (7) after realizing that for a smooth velocity field [92; 97]

$$\int \delta_a(\boldsymbol{r}_i - \boldsymbol{r})\boldsymbol{v}(\boldsymbol{r})\, d\boldsymbol{r} \approx \left[\boldsymbol{I} + \left(\int \frac{x^2}{2}\delta_a(x)\, dx\right)\nabla^2\right]\boldsymbol{v}(\boldsymbol{r})\Big|_{\boldsymbol{r}=\boldsymbol{r}_i} = (\boldsymbol{I} + \tfrac{1}{6}a_F^2\nabla^2)\boldsymbol{v}(\boldsymbol{r})\big|_{\boldsymbol{r}=\boldsymbol{r}_i},$$

where we assumed a spherical blob: $\delta_a(\boldsymbol{r}) \equiv \delta_a(r)$. We have defined here the "Faxén" radius of the blob $a_F \equiv (3\int x^2\delta_a(x)\, dx)^{1/2}$ through the second moment of the kernel function.

In multipole-expansion-based methods, the self-mobility of a body is treated separately by solving the single-body problem exactly (this is only possible for simple particle shapes). However, in the type of approach followed here the self-mobility $\boldsymbol{\mathcal{M}}_{ii}$ is also given by the same formula (21) with $i = j$ and does not need to be treated separately. In fact, the self-mobility of a particle in an unbounded three-dimensional domain *defines* the effective hydrodynamic radius a of a blob:

$$\boldsymbol{\mathcal{M}}_{ii} = \frac{1}{6\pi\eta a}\boldsymbol{I} = \eta^{-1}\int \delta_a(\boldsymbol{r}')\mathbb{O}(\boldsymbol{r}' - \boldsymbol{r}'')\delta_a(\boldsymbol{r}'')\, d\boldsymbol{r}'\, d\boldsymbol{r}'',$$

where the Oseen tensor $\mathbb{O}$ is given in (8). In general, $a_F \neq a$, but for a suitable choice of the kernel one can accomplish $a_F \approx a$ (for example, for a Gaussian $a/a_F = \sqrt{3/\pi}$ [97]) and thus accurately obtain the Faxén correction for a rigid sphere [33].

For an isotropic or tensor product kernel δ_a and an unbounded domain, the pairwise blob-blob mobility (21) will take the form

$$\boldsymbol{\mathcal{M}}_{ij} = f(r_{ij})\boldsymbol{\mathcal{I}} + g(r_{ij})\hat{\boldsymbol{r}}_{ij} \otimes \hat{\boldsymbol{r}}_{ij}, \tag{22}$$

where $\boldsymbol{r}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j$, and the hat denotes a unit vector. The functions of distance $f(r)$ and $g(r)$ depend on the specific kernel (and in the fully discrete setting on the spatial discretization of the Stokes equations) and will be different from those appearing in the RPY tensor (9). Nevertheless, as we will show numerically in Section 6.1, the functions f and g for our IB method are quite close in form to those appearing in

the RPY tensor. We note that the RPY tensor itself can be seen as a realization of (21) with the kernel being a surface delta function over a sphere of radius a [130].

We have demonstrated above that solving (20) is a way to apply the blob-blob mobility for a confined domain. In the method of regularized Stokeslets [2; 88; 28; 27] the mobility is obtained *analytically* by averaging the analytical Green's function with a kernel or envelope function specifically chosen to make the resulting integrals analytical. Note however that in that method the kernel δ_a appears only once inside the integral in (21) because only the force spreading is regularized but not the interpolation of the velocity; this leads to a nonsymmetric mobility matrix inconsistent with the Faxén formula (7). By contrast, our approach is guaranteed to lead to a symmetric-positive-semidefinite (SPD) mobility matrix $\mathcal{M}$, which is crucial when including thermal fluctuations [33; 3; 73].

3.2. ***Fully discrete formulation.*** To obtain a fully discrete formulation of the linear system (20) we need to spatially discretize the Stokes equations on a grid. The spatial discretization of the fluid equation used in this work uses a uniform Cartesian grid with grid spacing h, and is based on a second-order-accurate staggered-grid finite-volume (equivalently, finite-difference) discretization, in which vector-valued quantities such as velocity are represented on the faces of the Cartesian grid cells, while scalar-valued quantities such as pressure are represented at the centers of the grid cells [54; 53; 7; 71]. The viscous terms are discretized using a standard 7-point Laplacian (in three dimensions), accounting for boundary conditions using ghost cell extrapolation [53; 71].

Spreading and interpolation. In the fully discrete formulation of the fluid-body coupling, we replace spatial integrals in the semicontinuum formulation (20) by sums over fluid grid points. The regularized delta function kernel is discretized using a tensor product of one-dimensional immersed boundary kernels $\phi_a(x)$ of compact support, following Peskin [107]. To maximize translational and rotational invariance (i.e., improve grid-invariance) we use the smooth (three-times differentiable) 6-point kernel recently described by Bao et al. [9]. This kernel is more expensive than the traditional four-point kernel [107] because it increases the support of the kernel to $6^3 = 216$ grid points in three dimensions; however, this cost is justified because the new 6-point kernel improves the translational invariance by orders of magnitude compared to other standard IB kernel functions [9].

The interaction between the fluid and the rigid body is mediated through two crucial operations. The discrete velocity-interpolation operator $\mathcal{J}$ averages velocities on the staggered grid in the neighborhood of blob i via

$$(\mathcal{J}\boldsymbol{v})_i^\alpha = \sum_k v_k^\alpha \phi_a(\boldsymbol{r}_i - \boldsymbol{r}_k^\alpha),$$

where the sum is taken over faces k of the grid, α indexes coordinate directions (x, y, z) as a superscript, and $\boldsymbol{r}_k^\alpha$ is the position of the center of the grid face k in the direction α. The discrete force-spreading operator $\boldsymbol{\mathcal{S}}$ spreads forces from the blobs to the faces of the staggered grid via

$$(\boldsymbol{\mathcal{S}}\boldsymbol{\lambda})_k^\alpha = \Delta V^{-1} \sum_i \boldsymbol{\lambda}_i^\alpha \phi_a(\boldsymbol{r}_i - \boldsymbol{r}_k^\alpha), \tag{23}$$

where now the sum is over the blobs and $\Delta V = h^3$ is the volume of a grid cell. These operators are adjoint with respect to a suitably defined inner product, and the discrete matrices satisfy $\boldsymbol{\mathcal{J}} = \Delta V\, \boldsymbol{\mathcal{S}}^T$, which ensures conservation of energy [107]. Extensions of the basic interpolation and spreading operators to account for the presence of physical boundary conditions are described in Appendix D of [71].

We note that it is possible to change the effective hydrodynamic and Faxén radii of a blob by changing the kernel δ_a. Such flexibility in the kernel can be accomplished without compromising the required kernel properties postulated by Peskin [107] by using shifted or *split kernels* [7]:

$$\phi_{a,s}(\boldsymbol{q} - \boldsymbol{r}_k) = \frac{1}{2^d} \prod_{\alpha=1}^{d} \left\{ \phi_a\left[q_\alpha - (r_k)_\alpha - \tfrac{1}{2}s\right] + \phi_a\left[q_\alpha - (r_k)_\alpha + \tfrac{1}{2}s\right] \right\},$$

where s denotes a shift that parametrizes the kernel. By varying s in a certain range, for example, $0 \le s \le h$, one can smoothly increase the support of the kernel and thus increase the hydrodynamic radius of the blob by as much as a factor of 2. We do not use split kernels in this work but have found them to work as well as the unshifted kernels, while allowing increased flexibility in varying the grid spacing relative to the hydrodynamic radius of the particles.

Discrete constrained Stokes equations. Following spatial discretization, we obtain a finite-dimensional linear system of equations for the discrete velocities and pressures and the blob and body degrees of freedom. For the resistance problem, we obtain the rigidly constrained discrete Stokes system [71]

$$\begin{bmatrix} \boldsymbol{\mathcal{A}} & \boldsymbol{\mathcal{G}} & -\boldsymbol{\mathcal{S}} \\ -\boldsymbol{\mathcal{D}} & \boldsymbol{0} & \boldsymbol{0} \\ -\boldsymbol{\mathcal{J}} & \boldsymbol{0} & -\boldsymbol{\Omega} \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ \pi \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{g} = \boldsymbol{0} \\ \boldsymbol{h} = \boldsymbol{0} \\ \boldsymbol{w} = -\breve{\boldsymbol{u}} \end{bmatrix}, \tag{24}$$

where $\boldsymbol{\mathcal{G}}$ is the discrete (vector) gradient operator, $\boldsymbol{\mathcal{D}} = -\boldsymbol{\mathcal{G}}^T$ is the discrete (vector) divergence operator, and $\boldsymbol{\mathcal{A}} = -\eta \boldsymbol{\mathcal{L}}_v$ where $\boldsymbol{\mathcal{L}}_v$ is a discrete (vector) Laplacian; these finite-difference operators take into account the specified boundary conditions [53]. For impermeable bodies $\boldsymbol{\Omega} = \boldsymbol{0}$, which makes the linear system (24) a nested saddle-point problem in both Lagrange multipliers π and $\boldsymbol{\lambda}$. As explained in Appendix C, for permeable bodies $\boldsymbol{\Omega}$ is a diagonal matrix with $\Omega_{ii} = \kappa_p/(\eta \Delta V_i)$ for blob $i \in \mathcal{B}_p$,

where κ_p is the permeability of body p and ΔV_i is a volume associated with blob i. The right-hand side could include any external fluid-forcing terms, slip, inhomogeneous boundary conditions, etc. The system (24) can be made symmetric by excluding the volume weighting ΔV^{-1} in the spreading operator (23); this makes $\boldsymbol{\lambda}$ have units of force density rather than total force.

This nested saddle-point structure continues if one considers impermeable rigid bodies that are free to move, leading to the *discrete mobility problem*[1]

$$\begin{bmatrix} \mathcal{A} & \mathcal{G} & -\mathcal{S} & \mathbf{0} \\ -\mathcal{D} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathcal{J} & \mathbf{0} & \mathbf{0} & \mathcal{K} \\ \mathbf{0} & \mathbf{0} & \mathcal{K}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ \pi \\ \boldsymbol{\lambda} \\ \boldsymbol{U} \end{bmatrix} = \begin{bmatrix} \boldsymbol{g} \\ \boldsymbol{h} = \mathbf{0} \\ \boldsymbol{w} = -\breve{\boldsymbol{u}} \\ z = \boldsymbol{F} \end{bmatrix}. \tag{25}$$

After eliminating the velocity and pressure from this system, we obtain the saddle-point system (11) with the identification of the mobility with its discrete approximation

$$\mathcal{M} = \mathcal{J}\mathcal{L}^{-1}\mathcal{S} = \Delta V\, \mathcal{S}^T \mathcal{L}^{-1} \mathcal{S}, \tag{26}$$

which is SPD. Here $\mathcal{L}^{-1}$ is a discrete Stokes solution operator

$$\mathcal{L}^{-1} = \mathcal{A}^{-1} - \mathcal{A}^{-1}\mathcal{G}(\mathcal{D}\mathcal{A}^{-1}\mathcal{G})^{-1}\mathcal{D}\mathcal{A}^{-1}, \tag{27}$$

where we have assumed for now that $\mathcal{A}^{-1}$ is invertible; see [71] for the handling of periodic systems, for which the Laplacian is not invertible. Unlike for Green's-function-based methods, we never explicitly compute or form $\mathcal{L}^{-1}$ or $\mathcal{M}$; rather, we solve the Stokes velocity-pressure subsystems iteratively using the preconditioners described in [53; 20].

3.3. ***Preconditioning algorithm.*** In this subsection we describe how to solve the system (25) using an iterative solver, as we have implemented in the Immersed Boundary Adaptive Mesh Refinement software framework (IBAMR) [54]. Our codes are integrated into the public release of the IBAMR library. Note that the matrix-vector product is a straightforward and inexpensive application of finite-difference stencils on the fluid grid and summations over blobs. The key to an effective solver is the design of a good preconditioner, i.e., a good approximate solver for (25). The basic idea is to combine a preconditioner for the Stokes problem [42; 53; 20] with the indefinite preconditioner (18) with a block-diagonal approximation of the mobility $\widetilde{\mathcal{M}}$ constructed based on empirical fits of the blob-blob mobility, as we now explain in detail.

[1] Note that in actual codes it is better to use an increment formulation of the linear system where the unknowns are the changes of the unknowns from their values at the previous time step; this is particularly important when there is a nontrivial background flow to ensure that the (small) perturbative flows are resolved accurately.

Approximate blob-blob mobility matrix. A preconditioner for solving the resistance problem (24) was developed by some of the authors in [71]; readers interested in additional details should refer to this work. The preconditioner is based on approximating the blob-blob mobility with the functional form (22), where the functions $f(r)$ and $g(r)$ are obtained by fitting numerical data for the blob-blob mobility in an *unbounded* system (in practice, a large periodic system). This involves two important approximations, the validity of which only affects the *efficiency* of the linear solver but does *not* affect the *accuracy* of the method since the Krylov method will correct for the approximations. The first approximation comes from the fact that the true blob-blob mobility for the immersed boundary method is not perfectly translationally and rotationally invariant, so that the form (22) does not hold exactly. The second approximation is that the boundary conditions are not correctly taken into account when constructing the approximation of the mobility $\widetilde{\mathcal{M}}$. This approximation is crucial to the feasibility of our method and is much more severe, but as we will demonstrate numerically in Section 6, the Krylov solver converges in a reasonable number of iterations, correctly incorporating the boundary conditions in the solution.

The empirical fits of $f(r)$ and $g(r)$ are described in Appendix A of [71].[2] As we show in Section 6.1, these functions are quite similar to those appearing in the RPY tensor (9), and in fact, it is possible to use the RPY functions $f_{\mathrm{RPY}}(r)$ and $g_{\mathrm{RPY}}(r)$ in the preconditioner, with a value of the effective hydrodynamic radius a that depends on the choice of the kernel. Nevertheless, somewhat better performance is achieved by using the empirical fits for $f(r)$ and $g(r)$ developed in [71].

In [71], we considered general fluid-structure interaction problems over a range of Reynolds numbers, and constructed $\widetilde{\mathcal{M}}$ as a dense matrix of size $(dN_b) \times (dN_b)$, which was then factorized using dense linear algebra. This is infeasible for suspensions of many rigid bodies. In this work, we use the block-diagonal approximation (17) to the blob-blob mobility matrices, in which there is one block per rigid particle. Once $\widetilde{\mathcal{M}}$ is constructed and its diagonal blocks are factorized, the corresponding approximate body mobility matrix $\widetilde{\mathcal{N}}$ is easy to form, as discussed in more detail in Section 2.2. Note that these matrices and their factorizations need to be constructed only once at the beginning of the simulation, and can be reused throughout the simulation.

Fluid solver. A key component of solving the constrained Stokes problems (24) or (25) is an iterative solver for the unconstrained discrete Stokes subproblem

$$\begin{bmatrix} \mathcal{A} & \mathcal{G} \\ -\mathcal{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ \pi \end{bmatrix} = \begin{bmatrix} \boldsymbol{g} \\ \boldsymbol{h} \end{bmatrix},$$

[2] Code to evaluate the empirical fits is publicly available for a number of kernels constructed by Peskin and coworkers (3-, 4-, and 6-point) at http://cims.nyu.edu/~donev/src/MobilityFunctions.c.

for which a number of techniques have been developed in the finite-element context [42]. To solve this system, we can use GMRES with a preconditioner $\mathcal{P}_S^{-1}$ that assumes periodic boundary conditions so that the various finite-difference operators commute [41]. Specifically, the preconditioner for the Stokes system that we use in this work is based on a projection preconditioner developed by Griffith [53; 20]:

$$\mathcal{P}_S^{-1} = \begin{bmatrix} \mathcal{I} & h^2 \mathcal{G} \tilde{\mathcal{L}}_p^{-1} \\ \mathbf{0} & \eta \mathcal{I} \end{bmatrix} \begin{bmatrix} \mathcal{I} & \mathbf{0} \\ -\mathcal{D} & -\mathcal{I} \end{bmatrix} \begin{bmatrix} \eta^{-1} \tilde{\mathcal{L}}_v^{-1} & \mathbf{0} \\ \mathbf{0} & \mathcal{I} \end{bmatrix}, \tag{28}$$

where $\mathcal{L}_p = h^2(\mathcal{D}\mathcal{G})$ is the dimensionless pressure (scalar) Laplacian, and $\tilde{\mathcal{L}}_v^{-1} \approx (\mathcal{L}_v)^{-1}$ and $\tilde{\mathcal{L}}_p^{-1} \approx (\mathcal{L}_p)^{-1}$ denote approximate solvers obtained by a *single* V-cycle of a geometric multigrid method, as performed using the *hypre* library [43] in our IBAMR implementation. In this paper we will primarily report the options we have found to be best without listing all of the different combinations we have tried. For completeness, we note that we have tried the better-known lower- and upper-triangular preconditioners [42; 20] for the Stokes problem. While these simpler preconditioners are better when solving pure Stokes problems than the projection preconditioner (28) since they avoid the pressure multigrid application $\tilde{\mathcal{L}}_p^{-1}$, we have found them to perform much worse in the context of suspensions of rigid bodies. A possible explanation is that the projection preconditioner $\mathcal{P}_S^{-1}$ is the only one that is exact for periodic systems if exact subsolvers for the velocity and pressure subproblems are used.

Observe that one application of $\mathcal{P}_S^{-1}$ is relatively inexpensive and involves only $(d+1)$ scalar multigrid V-cycles. The number of iterations required for convergence depends strongly on the boundary conditions; fast convergence is obtained within 10–20 iterations for periodic systems, but as many as 100 GMRES iterations may be required for highly confined systems [20]. We emphasize that the performance of this preconditioner is highly dependent on the details of the staggered geometric multigrid method, which is not highly optimized in the *hypre* library, especially for domains of high aspect ratios such as narrow slit channels. For periodic boundary conditions, one can use FFTs to solve the Stokes problem, and this is likely to be more efficient than geometric multigrid especially because FFTs have been highly optimized for common hardware architectures. However, such an approach would require 3 scalar FFTs for *each* iteration of the iterative solver for the constrained Stokes problem (24) or (25), and this will in general be substantially more expensive than using only a few cycles of geometric multigrid as an *approximate* Stokes solver.

The use of an approximate Stokes solver instead of an exact one is an important difference between implementing the rigid multiblob method for periodic systems using the spectral Ewald method [90; 132] and our approach. The product of the blob-blob mobility with a vector can be computed more accurately and faster using the spectral Ewald method, in particular because one can adjust the cutoff

for splitting the computation between real and Fourier space arbitrarily, unlike in our method where the grid spacing is tied to the particle radius. However, for rigid multiblobs, one must solve the system (11), which requires potentially many matrix-vector products, i.e., many FFTs in the spectral Ewald approach. By contrast, in our method we solve the extended problem (25), and only solve the Stokes problems approximately using a few cycles of multigrid in each iteration. This will require more iterations but each iteration can be substantially cheaper than performing 3 FFTs each Krylov iteration. For nonperiodic systems, there is no equivalent of the spectral Ewald method, but see [58; 57] for some steps in this direction. Our method computes the hydrodynamic interactions in a confined geometry "on the fly" without ever actually computing the action of the Green's function exactly; rather, it is computed only approximately and the outer Krylov solver corrects for any approximations made in the preconditioner.

Preconditioning algorithm. We now have the necessary ingredients to compose a preconditioner for solving (25), i.e., to construct an approximate solver for this linear system. Each application of our preconditioner involves the following steps.

(1) Approximately solve the fluid subproblem

$$\begin{bmatrix} \mathcal{A} & \mathcal{G} \\ -\mathcal{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{v}} \\ \tilde{\pi} \end{bmatrix} = \begin{bmatrix} \boldsymbol{g} \\ \boldsymbol{h} \end{bmatrix}$$

using $N_s^{(1)}$ iterations of an iterative method with the preconditioner (28).

(2) Interpolate $\tilde{\boldsymbol{v}}$ to get the relative slip at each of the blobs, $\tilde{\boldsymbol{w}} = \mathcal{J}\tilde{\boldsymbol{v}} + \boldsymbol{w}$, and rotate the corresponding component from the current frame to the reference frame of each body.

(3) Approximately compute the unknown body kinematics $\boldsymbol{U}$.

   (a) Calculate $\tilde{\boldsymbol{\lambda}} = \widetilde{\mathcal{M}}^{-1}\tilde{\boldsymbol{w}}$ and rotate the result back to the fixed frame of reference. Here $\widetilde{\mathcal{M}}$ is a block-diagonal approximation to the blob-blob mobility matrix in the reference frame, as described on page 239; the factorization of the blocks of $\widetilde{\mathcal{M}}$ is performed once at the beginning of the simulation.

   (b) Calculate $\widetilde{\mathcal{F}} = \mathcal{F} + \mathcal{K}\tilde{\boldsymbol{\lambda}}$ and transform (rotate) $\widetilde{\mathcal{F}}$ to the body frame of reference.

   (c) Compute $\boldsymbol{U} = \widetilde{\mathcal{N}}\widetilde{\mathcal{F}}$ and transform it back to the fixed frame of reference, where $\widetilde{\mathcal{N}} = (\mathcal{K}\widetilde{\mathcal{M}}^{-1}\mathcal{K}^T)^{-1}$.

(4) Calculate the updated relative slip velocity at each of the blobs,

$$\Delta\boldsymbol{U} = \mathcal{K}^T\boldsymbol{U} - \tilde{\boldsymbol{w}},$$

and transform (rotate) it to reference body frame.

(5) Compute $\boldsymbol{\lambda} = \widetilde{\boldsymbol{\mathcal{M}}}^{-1}\,\Delta \boldsymbol{U}$ and transform $\boldsymbol{\lambda}$ back to the fixed frame of reference if necessary.

(6) Solve the corrected fluid subproblem to obtain the fluid velocity and pressure

$$\begin{bmatrix} \boldsymbol{\mathcal{A}} & \boldsymbol{\mathcal{G}} \\ -\boldsymbol{\mathcal{D}} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ \pi \end{bmatrix} = \begin{bmatrix} \boldsymbol{g} + \boldsymbol{\mathcal{S}}\boldsymbol{\lambda} \\ \boldsymbol{h} \end{bmatrix},$$

using $N_s^{(2)}$ iterations of an iterative method with the preconditioner (28).

A few comments are in order. The above preconditioner is not SPD so the outer Krylov solver should be a method such as GMRES or BiCGStab [116]. We prefer to use right-preconditioned Krylov solvers because in this case the residual computed by the iterative solver is the true residual (as opposed to the preconditioned residual for left preconditioning), and therefore, termination criteria ensure that the original system was solved to the desired target tolerance. We expect that the long-term recurrence GMRES method will require a smaller number of iterations than the short-term recurrence used in BiCGStab (but note that each iteration of BiCGStab requires *two* applications of the preconditioner). However, observe that GMRES can require substantially more memory since it requires storing a complete history of the iterative process.[3] This can be ameliorated by restarts at a cost of slowed convergence. If the iterative solver used for the Stokes solver in Steps (1) and (6) is a nonlinear method (most Krylov methods are nonlinear), then the outer solver must be a flexible method such as FGMRES. This flexibility typically increases the memory requirements of the iterative method (for example, it exactly doubles the number of stored intermediate vectors for FGMRES versus GMRES), and so an alternative is to use a linear method such as Richardson's method.[4] Note that when a preconditioned Krylov method is used for the Stokes subsolver, one additional application of the preconditioner is required to convert the system to preconditioned form for both left and right preconditioning, making the total number of applications of the Stokes preconditioner (28) $N_s^{(1)} + N_s^{(2)} + 2$ per Krylov iteration. By contrast, if Richardson's method is used in the Stokes subsolver, the number of preconditioner applications is $N_s^{(1)} + N_s^{(2)}$. Since in many practical cases the cost is dominated by the multigrid cycles, this difference can be important in the overall performance of the preconditioner. We will explore the performance of the preconditioner and the effect of the various choices in detail in Section 6.5.

[3] Each vector requires storing complete velocity and pressure fields, i.e., 4 floating-point numbers per grid cell, which can make the memory requirements of a GMRES-based solver with a large restart frequency quite high for large grid sizes.

[4] All of these iterative methods are available in the PETSc library [5] we use in our IBAMR implementation [54] of the above preconditioner, making it simple to try different combinations and study their effectiveness on any particular problem of interest.

4. Results: unbounded domain

In this section we investigate the accuracy of rigid multiblob models of spheres as a function of the number of blobs. We focus on spheres in an unbounded domain because of the availability of analytical results to which to compare, and not because the rigid multiblob method is particularly good for suspensions of spheres, for which there already exist a number of well-developed multipole expansion approaches. We also investigate the performance of the preconditioner developed in Section 2.2 for solving (11), for suspensions of spheres in an unbounded domain (e.g., clusters of colloids formed in a gel). For unbounded domains, we compute the product of the blob-blob mobility matrix $\mathcal{M}$ with a vector using the fast multipole method (FMM) developed specifically for the RPY tensor in [89]; this software makes four calls to the Poisson FMM implemented in the FMMLIB3D library[5] per matrix-vector product. As we will demonstrate empirically, the asymptotic cost of the rigid-multiblob method scales as $N_b \log N_b$, where N_b is the total number of blobs, with a coefficient that grows only weakly with density. We note that in this paper we use relatively tight tolerances ($\sim 10^{-9}$–10^{-8}) when computing the matrix-vector products solving the linear systems in order to test the robustness of the preconditioners; in practical applications much lower tolerances ($\sim 10^{-5}$–10^{-3}) would typically be employed, potentially lowering the overall computational effort considerably from what is reported here.

In this work, each sphere is discretized with n blobs of hydrodynamic radius a distributed on the surface of a sphere of *geometric* radius R_g. We discretize the surface of a sphere as a shell of blobs constructed by a recursive procedure suggested to us by Charles Peskin (private communication); the same procedure is used in [126]. We start with 12 blobs placed at the vertices of an icosahedron [129], which gives a uniform triangulation of a sphere by 20 triangular faces. Then, we place a new blob at the center of each edge and recursively subdivide each triangle into 4 smaller triangles, projecting the vertices back to the surface of the sphere along the way. Each subdivision approximately quadruples the number of vertices, with the k-th subdivision producing a model with $10 \cdot 4^{k-1} + 2$ blobs, leading to shells with 12, 42, 162, or 642 blobs; see Figure 2 in [34] for an illustration. In this section we study the optimal choice of a for a given resolution (number of blobs) and R_g.

An important concept that will be used heavily in the rest of this paper is that of an *effective hydrodynamic radius* $R_h \approx R_g + a/2$ of a blob model of a sphere (more generally, effective hydrodynamic extent). If we approach the rigid multiblob method from a boundary integral perspective, we would assign $R_h = R_g$ as the radius and treat the additional enlargement of the effective hydrodynamic radius as a numerical (quadrature + regularization) error. This is more or less how results

[5]The code is available at http://www.cims.nyu.edu/cmcl/fmm3dlib/fmm3dlib.html.

are presented in the recent work of Swan and Wang [126] (see for example their Figure 8), making the accuracy appear low even in the far field for a small number of blobs per sphere. However, we instead think of a rigid multiblob as an effective *model* of a sphere, whose hydrodynamic response mimics that of an equivalent sphere. A similar effect appears in lattice Boltzmann simulations, with a being related to the lattice spacing [84; 60]. To appreciate why it is imperative to use an effective radius, observe that even a single blob acts as an approximation of a sphere with radius $a > 0$. Similarly, one should not treat a line of blobs (see the leftmost panel in Figure 1) as a zero-thickness object (line); rather, such a line of rigidly connected blobs should be considered to model a rigid cylinder with finite thickness proportional to a [18]. We compute the effective hydrodynamic radius of our blob models of spheres next.

4.1. ***Effective hydrodynamic radii of rigid multiblob spheres.*** In this subsection we consider an isolated rigid multiblob sphere in an unbounded domain, and compute its response to an applied force f_p, an applied torque τ_p, and an applied linear shear flow with strain rate γ. Each of these defines an effective hydrodynamic radius by comparing to the analytical results for a sphere; therefore, each model of a sphere will have three distinct hydrodynamic radii.

The *translational radius* is measured from (see also [35])

$$R_h = \frac{f_p}{6\pi\eta u_p},$$

where u_p is the resulting sphere linear velocity, the *rotational radius* is (see also [35])

$$R_\tau = \left(\frac{\tau_p}{8\pi\eta\omega_p}\right)^{1/3},$$

where ω_p is the resulting angular velocity, and the effective stresslet radius is

$$R_s = \left(-\frac{3s_{11}}{20\pi\eta\gamma}\right)^{1/3}.$$

Here we compute the stresslet $\boldsymbol{s}$ induced on the rigid multiblob under an applied shear by setting an apparent slip $\breve{\boldsymbol{u}}_i = -\boldsymbol{v}(\boldsymbol{r}_i) = -\gamma(x, -y, 0)$ on blob i, and then solving the mobility problem to compute the constraint (rigidity) forces $\boldsymbol{\lambda}$. The stresslet $\boldsymbol{s}$ is the symmetric traceless component of the first moment of the constraint forces $\sum_{i\in\mathcal{B}_p} \boldsymbol{\lambda}_i \otimes \boldsymbol{r}_i$. In this work, we use R_h as the effective hydrodynamic radius when comparing to theory. This is because the translational mobility is controlled by the most long-ranged $1/r$ hydrodynamic interactions, and therefore, the far-field response of a rigid multiblob is controlled by R_h.

Observe that since we only account for translation of the blobs, only R_h is nonzero for a single blob, while R_τ and R_s are zero. Therefore, the minimal model of a sphere

Number of blobs	$a/s = \frac{1}{2}$			$a/s = \frac{1}{4}$		
	R_h/R_g	R_τ/R_g	R_s/R_g	R_h/R_g	R_τ/R_g	R_s/R_g
12	1.2625	1.2313	1.2461	1.0154	1.0292	0.9890
42	1.1220	1.1019	1.1316	1.0035	1.0147	0.9959
162	1.0530	1.0472	1.0567	0.9998	1.0073	0.9968
642	1.0239	1.0227	1.0250	0.9992	1.0036	0.9932
2562	1.0113	1.0111	1.0115	0.9994	1.0018	0.9986

Table 1. Effective translational, rotational, and stresslet hydrodynamic radii for rigid multiblob models of a sphere, for two choices of the blob-blob relative spacing a/s.

that allows for nontrivial rotlets and stresslets is the icosahedral model (12 blobs) [129]. Since the rigid multiblob models are able to exert a stress on the fluid they can change the viscosity of a suspension [129], unlike the single-blob models, which do not resist shear. It is important to note that the rigid multiblob models of a sphere are *not* perfectly rotationally invariant, especially for low resolutions. Therefore, the rigid multiblobs may exhibit a small translational velocity even in the absence of an applied force, or they may exhibit a small rotation even in the absence of an applied torque. In other words, the effective mobility matrix for a rigid multiblob model of a sphere can exhibit small off-diagonal components. Similarly, there will in general be small but nonzero components of the stresslet that would be identically zero for a perfect sphere. In general, we find these spurious components to be very small even for the minimally resolved icosahedral rigid multiblob.

A key parameter that we need to choose is how to relate the blob hydrodynamic radius a with the typical spacing between the blobs. Since our multiblob models of spheres are regular the minimal spacing between markers s is well-defined, and we expect that there will be some optimal ratio a/s that will make the rigid multiblob represent a true rigid sphere as best as possible. In a number of prior works the intuitive choice $a/s = \frac{1}{2}$ has been used, since this corresponds to the idea that the blobs act as a sphere of radius a and we would like them to touch the other blobs. However, as we explained above, it is not appropriate to think of blobs as spheres with a well-defined surface, and it is therefore important to study the optimal spacing more carefully.

In Table 1 we present the effective sphere radii obtained for different resolutions for two choices of a/s (we have investigated a broad range of spacings, not shown). The important observation we make is that even when the radii are far from geometric, such as for the 12-blob shell, the different radii are rather consistent with each other. This means that even low-resolution rigid multiblobs act like spheres as far as low-order moments (multipoles) are concerned. We note that the method developed in [35], in which rotational degrees of freedom are added to the blob description,

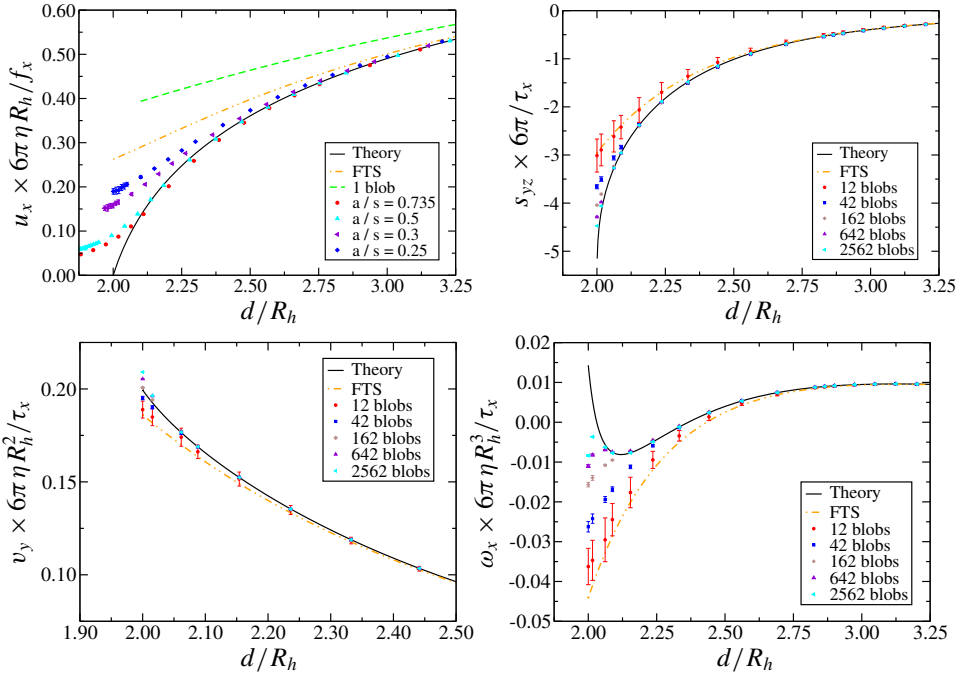


Figure 2. Hydrodynamic coupling between two identical spheres as a function of the center-to-center distance d. Twice the standard deviation as the two spheres are rotated relative to one another is shown as an error bar. Comparisons are made to Stokesian dynamics without lubrication corrections, i.e., truncation at the FTS level, and to "exact" theory [17; 67]; see legend. The top-left panel shows the average sphere velocity under the action of external unit forces $\boldsymbol{f}_1 = -\boldsymbol{f}_2 = \boldsymbol{f}$ directed along the line of collision, for a resolution of 162 blobs and for several values of a/s. The remaining three panels show nontrivial components of the pairwise mobility for a fixed $a/s = \frac{1}{2}$ and different resolutions (number of blobs per sphere; see legend). One sphere, located at $((d^2 - 4R_h^2)^{1/2}, 0, 2R_h)$, is subject to an external torque of magnitude 1 around the x-axis. The response of the second sphere located at the origin is measured: the top-right panel shows the stresslet $\boldsymbol{s}$ (i.e., the rotation-stresslet coupling), the bottom-left panel shows the linear velocity (i.e., the rotation-translation coupling) $\boldsymbol{v}$, and the bottom-right panel shows the angular velocity $\boldsymbol{\omega}$ (i.e., the rotation-rotation coupling) of the second sphere.

gives $R_h \approx R_\tau$ to within a fraction of a percent even for only 12 blobs per sphere. As the resolution is increased all hydrodynamic radii converge to the geometric radius R_g *linearly* with a/R_g (data not shown). The table also suggests that $a/s = \frac{1}{4}$ is better than $a/s = \frac{1}{2}$ because for $a/s = \frac{1}{4}$ all of the effective hydrodynamic radii are remarkably close to R_g even for the 12-blob model. However, as we show in the next subsection, the choice $a/s = \frac{1}{2}$ is substantially better when looking at how well lubrication forces are resolved between two spheres.

4.2. ***Mobility for a pair of spheres.*** To determine the best value of a/s in this subsection we examine the hydrodynamic interaction between two spheres as

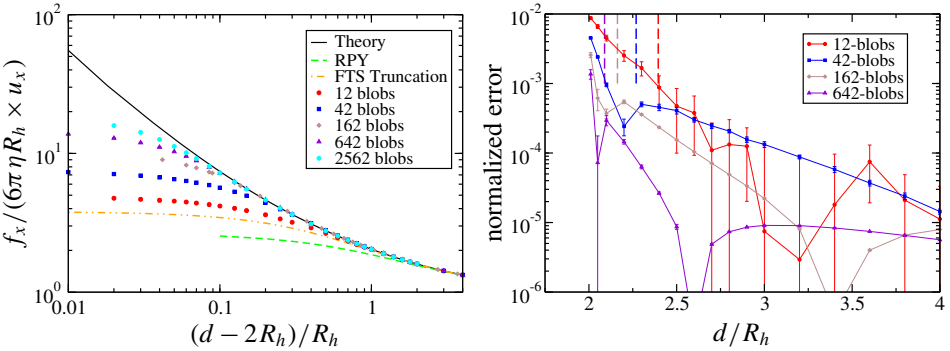


Figure 3. Lubrication forces on two identical spheres pulled toward each other with equal forces, for different resolutions (see legend). Twice the standard deviation as the two spheres are rotated relative to one another is shown as an error bar. Left: dimensionless normalized hydrodynamic resistance, i.e., the inverse of the hydrodynamic mobility shown in the top-left panel in Figure 2. Right: velocity error for one sphere with respect the exact theoretical result, normalized by the velocity at long distances ($f/6\pi\eta R_h$). The distance $d = 2(R_g + a)$ at which blobs start to overlap is marked as a vertical line of the same color as the corresponding symbols.

they approach each other. Since pairwise lubrication corrections are not added in our approach, it is important to investigate how well lubrication is resolved for different resolutions. To assert the accuracy of the rigid multiblob models we will examine several nontrivial components of the mobility between two spheres. Since rigid multiblob models are not rotationally invariant the exact value for the pair mobility depends on the relative orientation of the rigid multiblobs; here we report the mean and (twice the) standard deviation of the particle velocity as error bars, averaged over a sample of random orientations of the two particles. We note that we have compared our results to those obtained with the method developed in [35], where rotational DoFs are included in the blob description, and found only a small difference (not shown). This means that the inclusion of blob torques does not lead to an improvement in the accuracy with which pairwise hydrodynamic interactions are computed.

In our first test, we pull two spheres toward each other with equal but opposite forces directed along the line of collision. In the top-left panel of Figure 2 we compare the numerical results for spheres with 162 markers and different blob radii a with the exact result derived by Brenner [17]. One can see that for long distances all sensible choices of a provide a good agreement with the exact theory once the results have been scaled with the hydrodynamic radius R_h computed in Section 4.1. However, the choice of a makes a big difference at short distances. Specifically, for small a/s flow can "leak" in between the blobs and the lubrication force is substantially lowered. For large a/s, we expect that the corrugation of the effective hydrodynamic surface of the sphere will introduce deviations from a

spherical shape. As the figure illustrates, the best agreement between theory and numerical results is obtained for $a/s = \frac{1}{2}$. This intuitive choice has been used in other related methods [125; 100; 104], while the IB community has favored shorter distances between markers (but see the discussion in [71]). In the rest of the tests in this subsection we use $a/s = \frac{1}{2}$ and in the rest of the paper we will use $a/s \approx \frac{1}{2}$ unless otherwise noted.

We explore the "lubrication" forces between spheres at very close distances in more detail in Figure 3. In the left panel we show how the hydrodynamic force grows as the gap between the spheres decreases. The hydrodynamics of the low-resolution models start to deviate from the theory for gaps $\sim R_h/2$ or smaller, while the highest-resolution model (2562 blobs) shows a good agreement with the theory for gaps down to $0.1R_h$. The right panel of Figure 3 shows the velocity error for one of the spheres with respect the exact theoretical result. For all models the error is below 10^{-3} until distances where the blobs forming the spherical shells start to overlap, which is the intuitive distance above which we expect the rigid multiblob to act as a good approximation to a sphere.

In our second test, we measure the velocity of one sphere located at the origin when a second sphere, located at $(x, 0, 2R_h)$, is subject to an external torque applied around the x-axis. Since the Brenner theory is only valid for spheres approaching along the line of collision we use the expansion of Jeffrey and Onishi accurate to order $O(r^{-100})$ [67] to compare with our mobility results; this expansion is also used in Stokesian dynamics to compute near-field lubrication corrections for pairs of spheres and can be computed using the libStokes library of Ichiki [65]. One can see in the lower panels of Figure 2 that the low-resolution model (12 blobs) is similar to a Stokesian dynamics model that includes monopole (forces) and dipole (torques and stresslet) terms but no lubrication corrections. As the resolution of the models is increased the results agree better with the theory, as expected. Note, however, that the lack of lubrication corrections in our models prevents a perfect agreement down to contact distances. In the top-right panel of Figure 2 we compare the stresslet computed on the particle at the origin. Again, we observe that the 12-blob model is similar in accuracy (aside from the presence of nonzero error bars, i.e., variance) to the Stokesian dynamics method without lubrication corrections, while higher resolutions methods agree better with the theory, as expected.

4.3. ***Squirmer swimming speed.*** In this subsection we confirm the ability of our method to model an active sphere "squirmer" [32] with a prescribed tangential surface slip on the surface. This slip $\breve{\boldsymbol{u}}_i = \breve{\boldsymbol{u}}(\boldsymbol{r}_i)$ takes the following form in spherical coordinates:

$$\breve{\boldsymbol{u}}_r = 0, \qquad \breve{\boldsymbol{u}}_\phi = 0, \qquad \breve{\boldsymbol{u}}_\theta = B_1 \sin\theta.$$

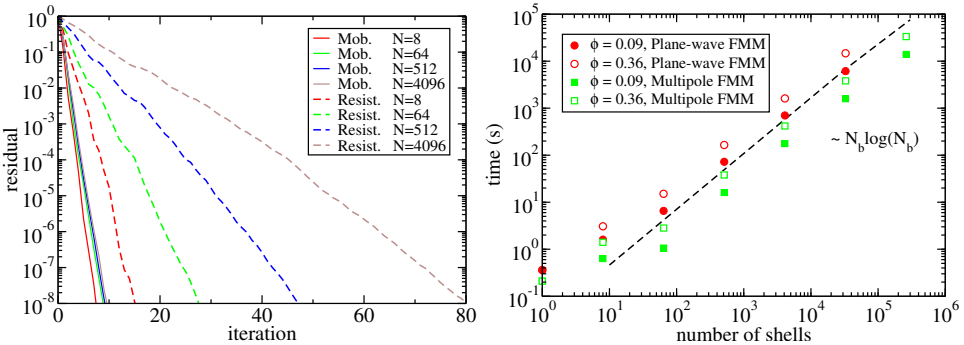


Figure 4. Left: convergence of preconditioned GMRES for the resistance and mobility problems for a finite subset of a cubic lattice of 42-blob spheres in an unbounded domain, for different numbers of particles, keeping the lattice spacing (closest distance between spheres) at $4R_g \sim 3.6R_h$ (corresponding to $\phi = 0.09$). Right: wall-clock time to solve the mobility problem with a tolerance 10^{-8} versus the number of spherical shells for two volume fractions ϕ, demonstrating the $O(N_b \log(N_b))$ asymptotic complexity, where N_b is the number of blobs. The matrix-vector product was computed with relative tolerance $0.5 \cdot 10^{-9}$ using the FMM method developed in [89]. We compare the performance of a parallel multipole-based FMMLIB3D code [51] with a more efficient but serial plane-wave FMM currently under development by the group of Leslie Greengard. Parallel runs used 8 cores and serial runs used a single core of an Intel Xeon 2.40 GHz processor.

The active translational velocity of the squirmer is well-known to be the surface average of the surface slip [121]

$$\boldsymbol{u} = -\langle \breve{\boldsymbol{u}} \rangle = \tfrac{2}{3} B_1 \hat{\boldsymbol{z}}.$$

We have numerically computed the swimming speed of a squirmer in an unbounded domain for different resolutions and compared to the theory. We obtain that the relative error ϵ in the swimming speed is linear in a/R_g, which is expected. However, the error has a large prefactor, $\epsilon \approx 3.5\, a/R_g$, which is not small for the low-resolution models. Furthermore, observe that linear convergence with the size of the blobs implies only order-$\frac{1}{2}$ convergence in the number of blobs since the number of blobs required to cover the surface of the sphere grows quadratically with the sphere radius.

These findings confirm that the rigid multiblob models converge to the correct swimming speed but the accuracy is not very good. This is, in fact, not so surprising because we did not include any adjustments to account for the (potentially large) difference between the effective hydrodynamic radius R_h and the geometric radius. That is, even though the effective no-slip surface has a radius $R_h > R_g$, we imposed the slip (in a locally averaged way) at the surface of a sphere of radius R_g. We will investigate these issues and potential ways to improve the accuracy with which

ϕ	Iterations 12 blobs	42 blobs
0.0014	4	4
0.011	5	6
0.09	9	10
0.18	13	13
0.36	20	23

Table 2. Iterations to solve the mobility problem with tolerance 10^{-8} for 4096 spheres discretized by 12 blobs, or for 512 spheres discretized by 42 blobs, arranged in a simple cubic lattice at different volume fractions ϕ; see Section 4.4.

active slip is imposed in future work. Here we simply note that rigid multiblob models are well-suited to qualitative studies of suspensions of many active particles. If one wishes to accurately model one or a few active particles higher-order methods such as boundary integral methods are preferable.

4.4. ***Suspension of spheres.*** In this subsection we study the convergence of the preconditioned Krylov solver for suspensions of many spheres. Our primary goal is to assess the effectiveness of our block-diagonal preconditioner for different packing densities (particle-particle distances) and numbers of particles. In the tests of this subsection we use spherical shells of 42 blobs subject to random forces, torques, and slips. We form a finite cubic subset of a simple cubic lattice and place it in an unbounded fluid domain. We use right-preconditioned GMRES without restarts, implemented using the PETSc [5] library.

First, we test the robustness with increasing system size, keeping the particles well-separated at a distance $4R_g \approx 3.6R_h$, which corresponds to volume fraction $\phi \approx 0.09$. The left panel of Figure 4 shows that the convergence is uniform and that the number of iterations to reduce the residual by a given factor depends very weakly on the number of spheres. This demonstrates the effectiveness of the block-diagonal preconditioner for the mobility problem. Next we investigate the robustness with respect to packing density. Table 2 shows the number of iterations to convergence for spheres arranged in a cubic simple lattice for several packing densities. When particles are far apart the solver converges fast because the hydrodynamic interactions between particles are weak and the preconditioner is designed to be an exact solver for a single body. As the spheres come closer together the preconditioner is not so effective and the Krylov solver needs to perform more iterations, as expected. However, even when the particles are relatively close the solver performs reasonably well. For example, when the spheres are at a distance of $2.55R_g \approx 2.3R_h$ ($\phi \approx 0.36$), the solver converges in 23 iterations. Of course, as the particles come closer and closer, and in particular, as the blobs on disjoint

spheres begin to overlap, we expect to see an increasing ill-conditioning of the linear system (11) and an increasing numbers of iterations. However, the rigid multiblob method should not be used in this regime, since it does not accurately resolve lubrication forces at such short distances. In the right panel of Figure 4 we show the wall-clock time to solve the mobility problem for different number of shells and volume fractions. Since the number of iterations is essentially independent of the system size we obtain a quasilinear scaling by using the FMM to compute the product between the blob-blob mobility matrix and a vector.

However, for the resistance problem explained in Section 2, the left panel of Figure 4 shows that the number of iterations to attain convergence increases with the number of particles as $N^{1/3}$, i.e., with the linear extent of the system. This is somewhat better than the $O(N^{1/2})$ iterations reported for Stokesian dynamics by Ichiki [65], but still much worse than the mobility problem.

We believe that this difference between resistance and mobility problems is physical rather than purely numerical. In particular, we expect that the same behavior will be observed in essentially any other iterative method, regardless of the specifics of the discretization of the problem (rigid multiblobs, boundary-integral methods, multipole expansions, regularized Stokeslets, etc.). To appreciate the difference between mobility and resistance problems, observe that it is possible to obtain a low accurate solution to the mobility problem by approximating each sphere by a single blob and then computing the matrix-vector product $\boldsymbol{\mathcal{M}}\boldsymbol{f} = \boldsymbol{u}$ using an FMM. On the other hand, to solve the resistance problem the linear system $\boldsymbol{\mathcal{M}}\boldsymbol{f} = \boldsymbol{u}$ has to be solved, which must account for the collective nature of hydrodynamic interactions. The difference appears because there is an effective far-field two-body approximation for the mobility $\boldsymbol{\mathcal{N}}$ (equivalently for $\boldsymbol{\mathcal{M}}$) but not for the resistance matrix $\boldsymbol{\mathcal{N}}^{-1}$ (or $\boldsymbol{\mathcal{M}}^{-1}$), which is essentially a multibody problem [40].

Mathematically, the difference appears because solving the saddle-point problem (11) is similar to computing the motion for force- and torque-free particles [115], even though forces and torques are applied on the particles. For force- and torque-free particles, the hydrodynamic fields and thus interactions with other particles decay faster than $1/r$. Therefore, the effective interactions that need to be captured by the iterative solver decay much faster for the mobility problem than for the resistance problem, making the former much easier. To confirm this intuition, we have studied (not shown) mixed resistance/mobility problems. When we fix the angular velocities but leave the linear velocities as free, we expect to see rapid convergence because the leading-order interactions that the Krylov method needs to capture decay as $1/r^3$. Indeed, we observe numerically that in this case the solver converges almost as well as for the pure mobility problem. However, when we fix the linear velocities of the spheres but let them freely rotate, we find that the solver converges almost as badly as for the pure resistance problem.

5. Results: single wall

In this section we study the accuracy of rigid multiblob models and the effectiveness of our block-diagonal preconditioner for particle suspensions sedimented near a single no-slip boundary. This is an important and common occurrence in practice, especially in the field of active matter, since many active particles have metallic components and are not density-matched with the solvent, and thus sediment to the bottom substrate. Some of the authors studied the diffusive dynamics of nonspherical particles near a no-slip boundary using a rigid multiblob approach in [34]. However, in that prior work, we only studied a single body, and therefore, all of the mobility matrices were simply formed as dense matrices. Here we explore in more detail the accuracy of rigid multiblob models and also demonstrate how to scale rigid multiblob computations to suspensions of thousands of rigid bodies.

To compute the hydrodynamic interactions between blobs in the presence of a single wall we use a pairwise approximation to the blobs' mobility which includes the effects of the wall in the Rotne–Prager tensor [122]. In our implementation, we compute the product of the blob-blob mobility matrix $\mathcal{M}$ with a vector using a direct $O(N_b^2)$ summation (here N_b is the number of blobs) implemented on a GPU using PyCUDA [78] in double precision; single precision can be used for lower accuracy requirements.[6] This is an ideal problem for using GPUs as an accelerator since the computation is trivially parallelized on shared memory. Furthermore, the communication requirements between the CPU and GPU are minimal, since only the positions of the blobs need to be communicated.[7] It is possible to implement a fast multipole method (FMM) for the RP(Y) tensor including wall corrections by using a system of images together with an FMM for unbounded domains [89; 51]. However, it is important to note that the asymptotically optimal FMMs on a CPU (even with multicore acceleration) will only be computationally more efficient than a direct sum on a GPU for more than about 100 000 blobs (in our testing on current hardware). Therefore, for many applications a simple GPU implementation is sufficient or even preferable over an asymptotically scalable implementation. Once a Rotne–Prager regularization of the construction of Gimbutas et al. [51] is developed and combined with an FMM, the asymptotic cost will be reduced to $O(N_b \log N_b)$ and our computations can be extended to millions of blobs.

In Section 5.1 we study the accuracy of rigid multiblob models for modeling a sphere close to a boundary, and in Section 5.2 we extend this study to a rigid cylinder (rod). In Section 5.3 we study the dynamics of a pair of active rods close to a no-slip boundary. In Section 5.4 we study the performance of our iterative solver on a

[6] Our codes are publicly available at https://github.com/stochasticHydroTools/RigidMultiblobsWall.

[7] For suspensions of identical bodies only the positions and orientations of the rigid bodies (so only up to 7 numbers per body) need to be communicated to the GPU.

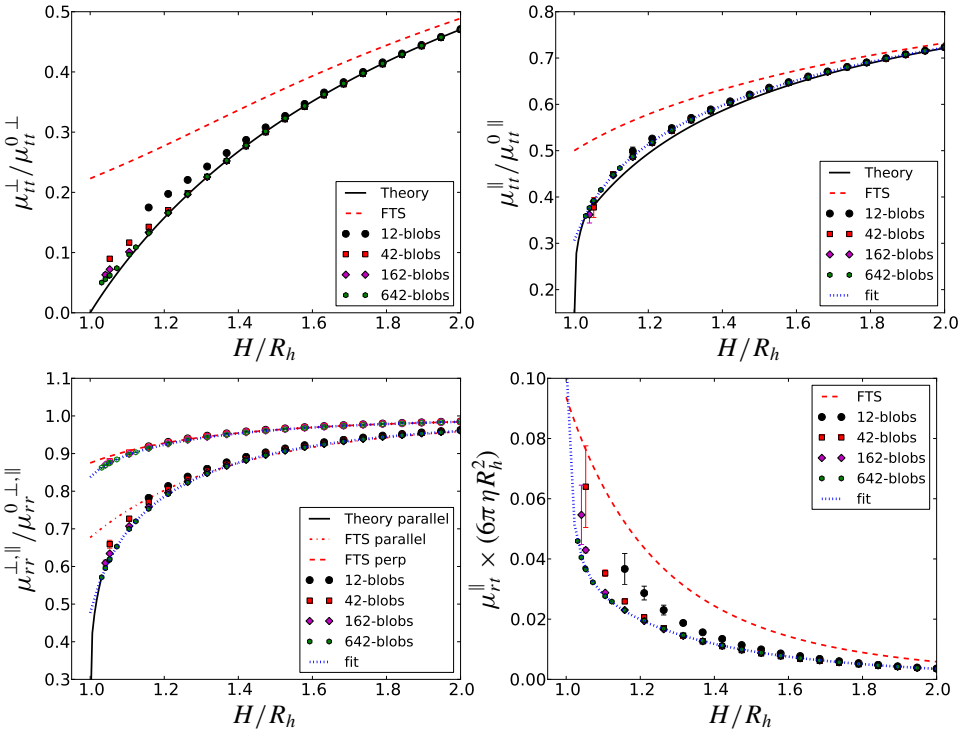


Figure 5. Selected components of the mobility $\boldsymbol{\mu}$ matrix for a sphere close to a wall, normalized by the corresponding bulk (unbounded domain) mobility $\boldsymbol{\mu}^0$ where possible. Comparison of the rigid multiblob results (symbols) is made to the best available theoretical results (solid black lines) and the FTS approximation used in Stokesian dynamics [122] (dashed and dashed-dotted lines). Empirical fits listed in Appendix D are shown with a dotted line. Top left: translational mobility $\mu_{tt}^{\perp}$ for a force applied perpendicular to the wall. Top right: translational mobility $\mu_{tt}^{\parallel}$ for a force parallel to the wall. Bottom left: rotational mobility for a torque applied parallel ($\mu_{rr}^{\parallel}$, filled symbols) or perpendicular ($\mu_{rr}^{\perp}$, empty symbols) to the wall. Bottom right: rotation-translation coupling mobility $\mu_{tr}^{\parallel}$ for force or torque parallel to the wall.

suspension of many rods, and demonstrate that the number of GMRES iterations is essentially independent of the number of particles just as for suspensions in an unbounded domain (see Section 4.4).

5.1. ***Sphere.*** In this section the mobility $\boldsymbol{\mu} \equiv \boldsymbol{\mathcal{N}}$ of a rigid multiblob sphere whose center is at a distance H from a no-slip boundary is compared with some theoretical results available in the literature. We use the shell models of spheres described in Section 4, and show the mean and standard deviation of the computed mobility averaged over a large number of random orientations of the rigid multiblob relative to the boundary. To denote the specific component of the mobility matrix we use a subscript tt for translational mobility, rr for rotational mobility, and tr for translation-rotation

coupling mobility, and we use a superscript $\perp$ or $\parallel$ to denote whether the direction of the force, torque, velocity, or angular velocity is perpendicular or parallel to the wall.

The top-left panel of Figure 5 presents the translational mobility of the sphere perpendicular to the wall together with the exact theory obtained by Brenner [17] (see also (D2) in [34] for a simple but accurate approximation). We also compare to the complete expression for the Rotne–Prager–Blake tensor derived by Swan and Brady [122], including stresslet corrections, which corresponds to an FTS truncation (plus degenerate quadrupole corrections) of the multipole hierarchy. It is evident that a single-blob model of a sphere, just like the substantially more complicated FTS truncation, does not recover the strong drop in the mobility (i.e., lubrication) at small distances to the wall. Rigid multiblob models do substantially better than the FTS truncation even with only 12 blobs (icosahedral multiblob), and as expected, the accuracy is improved with the addition of more blobs. As in Section 4, the numerical mobility never goes exactly to zero since we do not add lubrication corrections, and we expect the rigid multiblob model to only work well when the blobs do not overlap the boundary itself. In fact, we recall here that the RP tensor we use [122] does *not* include near-field corrections when blobs overlap the wall, and therefore, repulsive forces or other mechanisms should be used to ensure that the rigid multiblob is sufficiently far from the boundary. We empirically observe that the rotational invariance gets violated strongly if the gap to the wall is less than $2a/3$, which corresponds to 7% of the sphere radius for 12 blobs, and about 2% of the radius for the 642 blob model.

In the remaining panels of Figure 5 we investigate other components of the mobility. There are no closed-form expressions (even as infinite sums) for these components that are valid for all distances to the wall, so we use the best approximations available; see Appendix D in [34] for specific formulas. For the translational mobility parallel to the wall, shown in the top-right panel of the figure, we use a result based on lubrication theory [52] (see (D3) in [34]) when the sphere is very close to the wall ($H < 1.03R_h$), and an approximation to order $O((H/R_h)^5)$ for larger distances [44; 56] (see (D4) in [34]). It is clear that the rigid multiblob matches the theory for large distances but that the approximate theory is not very accurate for $H \lesssim 1.5R_h$ since the rigid multiblob results are clearly converging to something slightly different. As for the perpendicular mobility, we see that the icosahedral model (12 blobs) is substantially more accurate than an FTS truncation.

Our results for the rotational mobilities, for torque applied either perpendicular or parallel to the wall, are shown in the bottom-left panel of the figure and agree with the FTS results at large distances. We see slow but clear convergence of the rigid multiblob results for the translation-rotation coupling, shown in the bottom-right panel of the figure. This component of the mobility is therefore most difficult to capture accurately, as is evident from the fact that the FTS truncation does pretty

	$(\mu^0_{tt})_\parallel \times 4\pi\eta L$	$(\mu^0_{tt})_\perp \times 4\pi\eta L$	$(\mu^0_{rr})_\perp \times \pi\eta L^3/3$	$(\mu^0_{rr})_\parallel \times \pi\eta L^3/3$
14 blobs	3.422 (−0.53%)	2.612 (−0.23%)	1.216 (−0.31%)	n/a
86 blobs	3.324 (2.35%)	2.541 (2.48%)	1.240 (−2.34%)	11.564 (8.79%)
324 blobs	3.360 (1.29%)	2.588 (0.67%)	1.225 (−1.06%)	12.274 (3.19%)
∞ blobs	3.4040	2.6061	1.212	12.678

Table 3. Nontrivial elements of the bulk mobility matrix for empirically optimized rigid multiblob models of a cylinder of aspect ratio 6.35, shown in the three left panels of Figure 1. The value in the limit of infinite resolution is extrapolated numerically (see the main text) and reported in the last row. The percentages in parentheses correspond to the error relative to the infinite-resolution estimates.

poorly in this case. Since neither the FTS truncation nor the asymptotic lubrication results [52] are sufficiently accurate for comparison to experimental measurements, we have empirically fitted our highest-resolution results for the mobilities for which there are no exact theoretical expressions. We show the fits in Figure 5 and give details about the fits in Appendix D for the benefit of other researchers.

5.2. ***Cylinder.*** In this subsection, we consider a cylinder (rod) of length $L = 2.12$ and diameter $D = 2R = 0.325$ of aspect ratio $\alpha = L/D \approx 6.35$, mimicking the metallic rods studied in recent experiments [29], for three different levels of resolution. The minimal-resolution rigid multiblob has blobs placed in a row along the axis of the cylinder (a total of 14 blobs), while in the more resolved models, a hexagon (86 blobs) or a dodecagon (324 blobs) of blobs is placed along the circumference of the cylinder to better resolve it, as illustrated in Figure 1. We study different components of the mobility matrix $\boldsymbol{\mu} \equiv \boldsymbol{\mathcal{N}}$; to specify the direction of the force, torque, velocity, or angular velocity we use a subscript $\perp$ or $\parallel$ to denote whether the direction is perpendicular or parallel to the axes of the cylinder, respectively, and a superscript to denote whether the direction is perpendicular or parallel to the wall.

Bulk mobility. The first question that must be answered when constructing a rigid multiblob model of a given body is where to place the blobs and how to choose their hydrodynamic radius, to match the effective hydrodynamic response of the actual rigid body. Here we generalize the approach taken in Section 4.1 for spheres to a nonspherical body and show how to match the (passive) mobility of an actual rigid cylinder with a rigid multiblob model. Based on the results for spheres, for resolutions other than the minimally resolved model we cover the surface (similarly for the ends) of a cylinder uniformly with spheres keeping the spacing between blobs both around the circumference and the length of the cylinder uniform and fixed at $a/s = \frac{1}{2}$.

Because there are no exact analytical results for the mobility of a cylinder even in an unbounded domain, we estimate the true mobility of the cylinder $\boldsymbol{\mu}^0$ in an

unbounded domain numerically. Specifically, we place the blobs on the surface of a cylinder of length L and radius R (i.e., the true geometric surface of the actual rod we are modeling), and numerically compute the mobility for different resolutions. We then extrapolate to the limit $a \to 0$ using the two finest resolutions (86 and 324 blobs) based on our knowledge that the error is linear in a. For the translation-translation mobility we obtain $(\mu^0_{tt})_{\parallel} \times 4\pi\eta L = 3.404$ (compare to 3.295 from slender-body theory [76]) and $(\mu^0_{tt})_{\perp} \times 4\pi\eta L = 2.606$ (compare to 2.619 from slender-body theory [76]), while for the rotation-rotation mobility we get $(\mu^0_{rr})_{\perp} \times \pi\eta L^3/3 = 1.212$ (compare to 1.211 from slender-body theory [18]) and $(\mu^0_{rr})_{\parallel} \times \pi\eta L^3/3 = 12.678$; see the last row in Table 3.

Our goal here is to match the bulk mobility $\boldsymbol{\mu}^0$ of our rigid multiblob models to that of a true cylinder as closely as possible. To do this, for the surface-resolved models (86 and 324 blobs), we place the blobs on the surface of a cylinder of the *same* aspect ratio $\alpha = 6.35$ but with the geometric radius R_g of this cylinder allowed to be smaller than the geometric radius of the actual particle, while keeping the blob spacing $a/s = \frac{1}{2}$. We then numerically optimize the value of R_g to minimize a measure of the error with respect to (extrapolated) mobility of a true cylinder, to obtain $R_g/R = 0.90$ for the 86-blob model and $R_g/R = 0.95$ for the 324-blob model. For the minimally resolved model, we empirically tune both the geometric length (i.e., the distance between the centers of the two furthest blobs) to $L_g = 0.914L$ and the blob radius to $a = 1.103R$ while keeping the number of blobs fixed at $N_b = L/R + 1 = 14$ as suggested by Bringley and Peskin [18]. Table 3 shows the resulting infinite-domain mobilities for each resolution along with the relative error compared to the extrapolated values for infinite resolution. We see a relative error always less than 2.5% even for the minimally resolved model, except for rotation of the cylinder around its own axis; recall that the minimally resolved model cannot support a torque around the axis of the cylinder.

Mobility for a sedimented rod. Having determined the geometric parameters for the rigid multiblob models based on motion in an unbounded domain, we now study the accuracy of the three different resolutions for a cylinder close to a no-slip boundary. We assume that the cylinder is parallel to the wall with the centerline of the rod at a distance H from the no-slip boundary.

Figure 6 compares the computed mobility coefficients to available theoretical and experimental results. As could be expected, the decrease in mobility when approaching the boundary is clearly underestimated with the minimally resolved model. The left panel of the figure shows the translational mobilities. For $\mu^{\perp}_{\perp}$, the higher resolutions are in good agreement with the experimental measurements of Trahan and Hussey for a sedimenting rod with aspect ratio $\alpha = 5.05$ [128]. Our numerical results also match well the theory of Jeffrey and Onishi [66] for an infinite cylinder when $H < 2R$. It is important to emphasize that our model is significantly

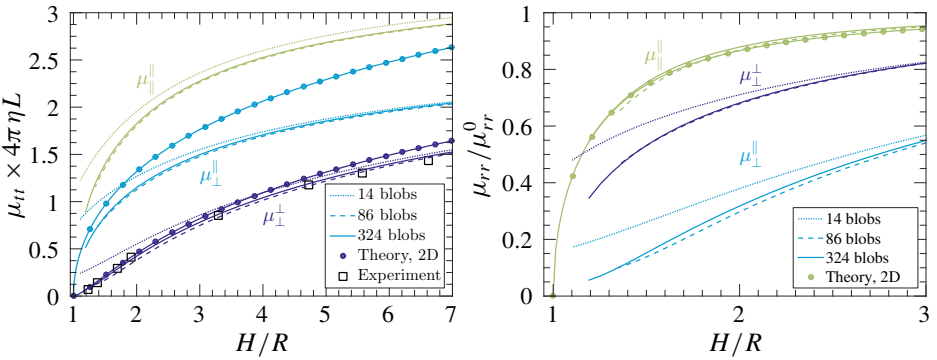


Figure 6. Mobility coefficients for a cylinder of aspect ratio $\alpha = L/(2R) = 6.35$ when it is parallel to the wall, as a function of the height of the rod centerline H/R. The superscript of the mobility denotes the direction with respect to the wall, while the subscript denotes the direction with respect to the rod axis. Left: translational mobility coefficients $\boldsymbol{\mu}_{tt}$ normalized by $4\pi\eta L$ as in [128]. The curves with circles correspond to the formulas from Jeffrey and Onishi for an infinite cylinder near a wall [66]. The black squares correspond to the experimental measurements of Trahan and Hussey for a rod with aspect ratio $\alpha = 5.05$ [128]. Right: rotational mobility coefficients $\boldsymbol{\mu}_{rr}$ normalized by the corresponding bulk value. The curve with circles corresponds to an infinite cylinder near a wall [66].

more accurate than slender-body theory near boundaries; the slender-body theory results from [72; 98; 76] (not shown here) are reasonably accurate only when $H/R > 3.5$ for aspect ratios $\alpha > 9$ [128]. The rotational mobilities of the rod are shown in the right panel of Figure 6. For the rotational mobility $\mu_{\parallel}^{\parallel}$, all three resolutions are in good agreement with the theory of Jeffrey and Onishi [66] for an infinite cylinder. For $\mu_{\perp}^{\parallel}$ and $\mu_{\perp}^{\perp}$, the minimally resolved model shows substantial errors near the wall, but the two higher-resolution models agree with each other quite well over a broad range of distances.

5.3. ***Active rod pair.*** In this subsection we apply the rigid multiblob method to a problem of recent experimental and theoretical interest: the dynamics of a pair of active "nanorods" that exhibit a "pusher" or extensile dipolar flow at large distances. Specifically, we compute the motion of a dimer of tripartite nanorods, as studied in recent experiments by Davies Wykes et al. [29]. The rods have diameter $0.325\,\mu$m and length $2.12\,\mu$m, and are in force and torque equilibrium (under the action of gravity and van der Waals and electrostatic interaction forces with the boundary) at some distance from the wall that has not been measured in the experiments. The rods are constructed of three metal sections, in the arrangement gold-platinum-gold and create a dipolar extensile (pusher) far-field flow. As such, they do not propel themselves in isolation but experiments show the formation of dimers of rods that actively rotate in a direction that is opposite of that predicted by recent simulations [106], which neglect the presence of the bottom wall. In agreement with the

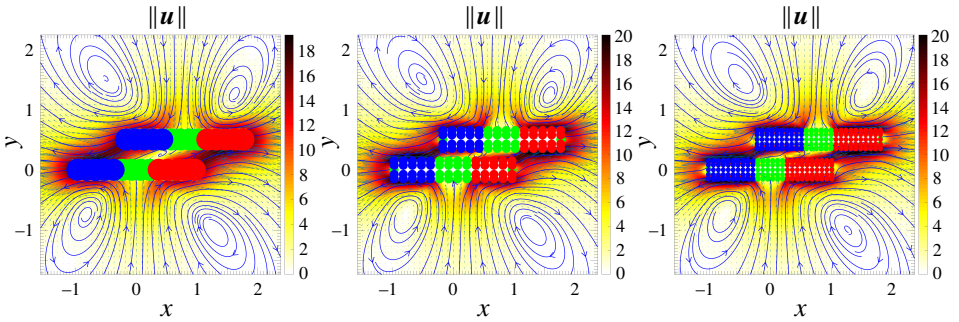


Figure 7. Active flow around a pair of extensile nanorods composed of three segments (shown with blue, green, and red) sedimented on top of a no-slip boundary (the plane of the image) and viewed from above for the three different levels of resolution illustrated in Figure 1: minimally resolved (left), marginally resolved (middle), and well-resolved (right). The colored disks (red, blue, or green) are projections of the blobs, with no-slip conditions on the green blobs and active slip of magnitude 20 μm/s on the blue and red blobs, directed away from the green segment. A cut through the flow field is shown in the lab frame as a vector field along with streamlines, with the magnitude of the velocity shown as a color scale plot.

experiments, our simulations show the formation of a stable rod pair that touch each other tangentially and rotate (without exhibiting a significant translation) around an axis perpendicular to the wall in a direction consistent with the experimental measurements.

The exact details of the active flows near the surface of the rods have not been measured experimentally and are difficult to predict analytically because this requires resolving the thin slip boundary layer (of thickness related to the Debye length) around the rods, as well as the knowledge of a number of material constants that are not known accurately. To obtain a qualitative understanding of the dynamics of the rods we impose an apparent tangential surface slip velocity on the two gold sections, directed away from platinum center and having a magnitude of 20 μm/s; no slip is imposed on the platinum section. Note that both gravity and the active slip pull the rods toward the wall, so we use an ad hoc repulsive force with the wall to balance the distance between the cylinder centerline and the wall at one cylinder diameter. Due to electrostatic interactions, a stacking of the two rods with the gold end of one rod aligned with the platinum center of the other is observed experimentally [29]; here we study the flow around such a pair of aligned rods.

In Figure 7 we show the instantaneous flow around a dimer of active rods, as computed using the procedure described in Appendix B and seen from above, for three different resolutions: a minimally resolved, a moderately resolved, and a well-resolved model. The imposition of the slip at the surface of the blobs becomes more and more accurate as the resolution is improved; however, we see a rather good match between the three flow fields even relatively close to the rods and wall.

Our simulations, which correctly take into account the physical boundary conditions, estimate the angular frequency of rotation of the dimer to be approximately 0.64 Hz in the counterclockwise direction, consistent with experimental observations [29]. The estimated angular velocities are 0.62, 0.67, and 0.63 Hz for each resolution, respectively. We will study the dynamics of active nanorods near a no-slip surface in more detail in the future; in the next subsection we demonstrate that the calculations above can be scaled to suspensions of thousands of rods.

5.4. ***Suspension of rods near a boundary.*** In this subsection we test the efficiency of the preconditioner outlined in Section 2.2 on a suspension of active rods sedimented near a wall. We have already seen that the block-diagonal preconditioner is able to account for the hydrodynamic interactions among the particles in a modest number of iterations for unbounded flow. Here we show that this continues to hold even when the wall strongly dominates the hydrodynamics, and investigate how important it is for the preconditioner to know about the presence of the boundary. Namely, recall that in the block-diagonal preconditioner the diagonal blocks of $\widetilde{\mathcal{M}}$ correspond to the blob-blob mobility for an individual rod *in the presence of the boundary*. Since $\widetilde{\mathcal{M}}$ depends on the configuration of *each* rod relative to the wall, unlike for an unbounded suspension, all diagonal blocks need to be factorized anew for each new configuration. However, it is also possible to use an *approximate* block-diagonal preconditioner that assumes an *unbounded* suspension, i.e., neglects the presence of the boundary when computing a block-diagonal approximation of the blob-blob mobility. This seems like a strong approximation to be made for objects close to a no-slip wall; however, the investigations below will demonstrate that the Krylov solver can account not only for the rod-rod interactions, but also for the rod-wall interactions. This is an important finding because we recall that in the Green's-function-free method described for confined suspensions in Section 3, the boundary conditions are completely ignored in the preconditioner.

In these tests we discretize cylinders of aspect ratio $\alpha = L/D \approx 6.4$ either by placing 98 blobs on the surface of a cylinder of geometric length $L_g = L$ and geometric radius $R_g = 0.863R$, keeping $a/s = \frac{1}{2}$, or by placing 21 blobs of radius $a = 1.02R$ uniformly spaced along a line segment of length L. For testing purposes, we generate random periodic packings of N_r rectangles at a surface packing density ϕ_a using a molecular dynamics code [36]. We then use these hard-rectangle packings to generate a configuration of nonoverlapping cylinders that are parallel to the wall and at a constant distance H from the wall; we do not expect to see different results if some randomness is added to the heights and orientations of the cylinders relative to the wall, as long as their surfaces remain sufficiently far from the wall. In our tests we vary the centerline height H from $H = 0.75D$ to $H = 2D$, the area fraction ϕ_a from 0.01 to 0.6, and the number of rods N_r from 10 to 10^4; the number of blobs

ϕ_a	Resolution	Wall-corrected	Unbounded
0.01	21	12	17
	98	16	28
0.1	21	19	23
	98	22	32
0.2	21	20	25
	98	23	34
0.4	21	25	29
	98	27	33
0.6	21	30	33
	98	31	43

Table 4. Number of GMRES iterations required to reduce the residual by a factor of 10^8 for several surface packing fractions and two different resolutions (number of blobs per rod), for $H/D = 0.75$ and $N_r = 1000$ rods. The full block-diagonal preconditioner, which takes into account the wall corrections for each body, is compared to the approximate preconditioner, in which all wall corrections are neglected.

varies in the range from $N_b = 200$ to about $N_b = 10^6$. For our implementation and hardware (a Tesla K20 GPU) one GMRES iteration takes around 0.3 s for $N_b = 10^4$, 1 s for $N_b \approx 2 \cdot 10^4$, 20 s for $N_b \approx 10^5$, and 220 s for $N_b \approx 3 \cdot 10^5$; we emphasize again that by using an FMM one can change the scaling from $O(N_b^2)$ to $O(N_b \log N_b)$ and thus substantially reduce the computational times for large N_b; see the right panel of Figure 4. To ensure a nontrivial right-hand side of the linear system when testing the iterative solver, each blob is prescribed a random slip velocity and a random force, producing a random force and torque on each cylinder.

Table 4 shows the scaling of the number of GMRES iterations with the area fraction ϕ_a for a fixed number of rods $N_r = 1000$, and compares the efficiency of the preconditioner using the full (wall-corrected) to that using the approximate (no wall contributions) block-diagonal preconditioner. We observe that the number of iterations increases slowly with the area fraction for both resolutions and reaches a maximum of 31 iterations for $\phi_a = 0.6$ with the wall-corrected preconditioner. Therefore, as we already saw in Section 4.4, the performance of the preconditioner is not highly sensitive to near-field interactions. When using the approximate block-diagonal preconditioner without the wall corrections, the number of iterations is increased, as expected. However, this increase never exceeds 50%, which means that even a poor approximation of the mobility can be used in the preconditioner in practice. Table 5 shows the scaling of the preconditioner with the number of rods N_r for a fixed area fraction $\phi_a = 0.1$. The number of iterations rapidly converges to around 20 and becomes independent of the number of rods for both resolutions

N_r	Resolution	$H/D = 0.75$ Iterations	$H/D = 0.75$ Time (s)	$H/D = 2$ Iterations	$H/D = 2$ Time (s)
10	21	7	0.15	7	0.15
	98	8	1.49	9	1.51
100	21	14	1.95	13	1.52
	98	19	18.9	18	35.6
1 000	21	19	32.7	16	29.8
	98	22	620	20	559
5 000	21	18	520	16	4 500
	98	23	10 200	22	12 400
10 000	21	20	2 050	17	1 430
	98	23	39 400	21	36 300

Table 5. Number of iterations and wall-clock time (using a direct GPU matrix-vector product on a Tesla K20 GPU) to solve the mobility problem with tolerance 10^{-8} using the wall-corrected preconditioner at $\phi_a = 0.1$, for different numbers of rods and proximities of the rods to the wall.

and heights. This confirms that the results obtained for a suspension of spheres in Section 4.4 apply to confined suspensions as well. Note that for the largest system sizes studied in Table 5 a linear-scaling FMM implementation would likely be substantially more efficient than the quadratic-scaling GPU implementation employed here.

6. Results: confined domains

In this section, we numerically explore the accuracy and efficiency of the rigid multiblob immersed boundary (IB) method described in Section 3. This method is suited to fully confined (bounded) domains, and here we model suspensions of spheres in a periodic domain, a slit channel (i.e., two parallel walls), and a square (duct) channel. As discussed in more detail in Section 3, for periodic suspensions it is possible to use FFT-based methods [90; 132; 73; 31] to obtain the product of the blob-blob mobility matrix with a vector. Future work should compare the method developed here with such approaches, especially for Brownian suspensions.

The effective hydrodynamic radius of an IB blob (also called a "marker" or "IB point" in the IB literature [107]) can be computed numerically by dragging a single blob with a constant applied force through a large periodic grid with spacing h and applying the Hasimoto periodic correction [7; 33]. When averaged over many positions of the marker relative to the underlying grid, for the 6-point kernel [9] used

here[8] we obtain $a \approx 1.47\,h$. The geometry of the rigid multiblob models of a sphere used here is the same as in Sections 4 and 5. We also know from Section 4.2 that the spacing between the blobs should be around $s \approx 2a \approx 3h$, which is somewhat larger than the spacing $s \approx 2h$ used in [71], and leads to improved conditioning of the blob-blob mobility matrix. In fact, we observe that when distinct blobs overlap, the preconditioned GMRES solver described in Section 3.3 shows significantly worse performance than when the blobs are not overlapping (or just touching). We therefore recommend using $s \gtrsim 3h$ for rigid multiblob suspensions at zero Reynolds numbers.

Determining the exact spacing is somewhat of an art and is problem-specific. In the IB approach developed here, the fast multipole method used in Section 4 and the GPU matrix-vector product used in Section 5 are replaced by a geometric multigrid method, which works best for grid sizes that are powers of 2. Once the exact spacing is determined, the effective hydrodynamic radius of the rigid multiblob can be determined numerically; we get very similar results for the IB method to those for an unbounded domain in Section 4.1, after setting $a = 1.47\,h$. For confined suspensions, the ratio of the size of the particles to the domain size is typically fixed to some experimental value, and this constrains the choice of the number of grid cells and spacing between the blobs. In all of the tests presented here, we have empirically optimized the appropriate value for the grid size and the spacing s in the interval $2h$ to $3h$, and report the chosen values. As explained earlier, it is possible to use split IB kernels to gain more flexibility in choosing the grid sizes and blob spacing.

In Section 6.1, we investigate in more detail the loss of perfect translational and rotational invariance of the blob-blob mobility and the mobility of rigid multiblob spheres, and demonstrate that by using the improved 6-point kernel [9] our method is able to minimize the grid artifacts to a significant extent. Note, however, that there is an additional loss of rotational invariance for rigid multiblob models that comes from discretizing the bodies using blobs; this unphysical bias exists even in the absence of a fluid grid. In Section 6.2 we explore in more detail the accuracy for different resolutions for a periodic suspension of spheres by comparing to reference results from multipole-based methods. In Section 6.3 we compute the mobility of a sphere in a slit channel and compare to existing theories for a number of resolutions. In Section 6.4 we compute the effective quasi-two-dimensional diffusion coefficient of a boomerang colloid in a slit channel, and compare to recent experimental measurements [21; 22; 23]. In Section 6.5 we optimize the convergence of the iterative linear solver for suspensions of many bodies, and demonstrate that the number of GMRES iterations is essentially independent of the number of particles, even in confined domains such as slit channels. Finally, we study the sedimentation velocities in a bidisperse suspension of spheres in Section 6.6, and compare to

[8] As summarized in [7; 33], $a \approx 1.25h$ for the widely used 4-point kernel [107], and $a \approx 0.91h$ for the 3-point kernel [112].

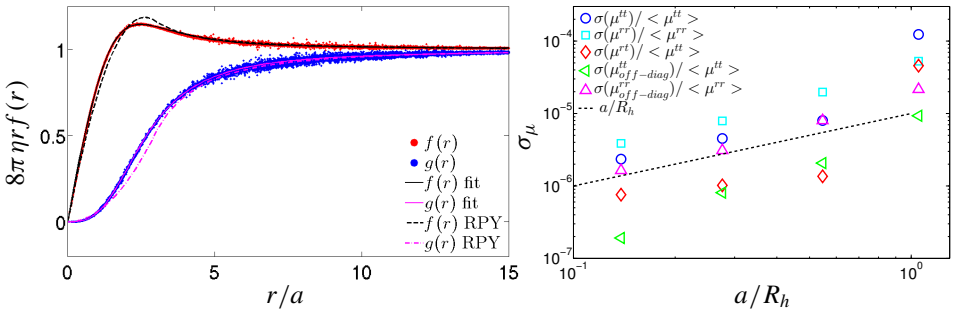


Figure 8. Translational and rotational invariance of the rigid multiblob IB method. Left: empirical values of the blob-blob mobility functions $f(r)$ and $g(r)$ appearing in (22), normalized by $8\pi\eta r$ so that they asymptote to unity. An empirical fit through the data is compared to the RPY tensor for blob radius $a = 1.47h$. Note that the scatter around the fit for $r \gtrsim 5a$ is dominated by periodic artifacts due to the finite size of the grid. Right: standard deviations of the diagonal (translation-translation and rotation-rotation) and cross-coupling (rotation-translation) components of the mobility matrix for spheres discretized with 12, 42, 162, or 642 markers shells (i.e., for decreasing a/R_h). Also shown is the typical magnitude of the off-diagonal components of the translation-translation and rotation-rotation mobility matrices, which should be zero for a perfect sphere.

recent Stokesian dynamics simulations [131; 132]. In Appendix C we study flows around permeable rigid bodies.

6.1. ***Translational invariance.*** As explained in detail in Section 3, for sufficiently large domains the blob-blob mobility computed by the IB method has the approximate form (22). Deviations from this formula arise because of the imperfect translational and rotational invariance due to grid artifacts. The two functions $f(r)$ and $g(r)$ are expected to be similar to those appearing in the RPY tensor (9). We obtain the actual form of the functions $f(r)$ and $g(r)$ empirically by fitting numerical data for the parallel and perpendicular mobilities of a pair of blobs placed in a large periodic domain; see [71] for more details. The results are shown in the left panel of Figure 8, and are compared to the RPY tensor for spheres of radius $a = 1.47h$. We have empirically fitted the numerical results for $f(r)$ and $g(r)$ with a fit that has the proper asymptotic behavior at large and short distances; see Appendix A in [71] for more details. This fit is used in the preconditioner as an analytical approximation of the diagonal blocks of the blob-blob mobility matrix. We see that the differences between the fits and the RPY tensor are rather small, and also confirm the improved translational invariance of the 6-point kernel [9] as evidenced in the small scatter of the points around the fits. This confirms our expectation that the rigid multiblob IB method will behave similarly to an RPY-based method in terms of accuracy.

In the right panel of Figure 8 we investigate the translational and rotational invariance of rigid multiblob models of spheres as a function of the resolution. We randomly move and rotate a sphere relative to the underlying grid and compute the

elements of the mobility matrix. The mean of these elements defines an effective translational and rotational radius consistent with the results presented in Section 4.1 (not shown). In the figure we show the normalized standard deviation of the elements of the mobility matrix as a function of the resolution (number of blobs). As expected, the normalized standard deviation decreases linearly with the size of the blobs a (equivalently, the inverse of the square root of N_b), and is below 10^{-4} for all mobility elements even for the 12-blob (icosahedral) model [129].

6.2. ***Periodic suspension of spheres.*** In this section we apply our rigid multiblob method to a benchmark resistance problem in a periodic suspension of 108 spheres moving with random linear and angular velocities. This benchmark was developed by Anthony Ladd, who supplied us with a random and a face-centered cubic (FCC) configuration of spheres, at a low density of $\phi = 0.05$, as well as a high density of $\phi = 0.45$. He also supplied us with the results for the resulting forces and torques obtained using the HYDROMULTIPOLE code [24]. Note that pairwise lubrication corrections have been included in the multipole expansion method used for these calculations [24; 81; 82]; to our knowledge no method has accounted for three-body lubrication corrections.

The rigid multiblob models used in this study have been chosen to give a blob spacing close to $s \approx 2a \approx 3h$, while ensuring that the number of grid cells is integer given the specified unit cell length in the benchmark configurations, and to have a specified effective hydrodynamic radius $R_h \approx 1$ for 5 different resolutions: a single blob per sphere (similar to a truncation with only one monopole per sphere), 12 blobs (geometric radius $R_g = 0.7896$ and grid spacing $h = 0.2778$), 42 blobs ($R_g = 0.8899$ and $h = 0.1667$), 162 blobs ($R_g = 0.9502$ and $h = 0.08929$), and 642 blobs ($R_g = 0.9766$ and $h = 0.0463$) per sphere. The results for the x component of the computed forces on the spheres are illustrated in Figure 9; similar results are observed for other components.

In the left panel of Figure 9, we focus on the low-density suspension ($\phi = 0.05$) and compare the forces computed by the rigid multigrid method with those computed using $L = 8$ multipole moments, as well as the results of the Stokesian dynamics (SD) code of Ichiki [65] (which roughly corresponds to $L = 2$ moments). The overall agreement is quite good, but notice that even with 162 blobs per sphere we do not resolve the lubrication force between particles numbered 48 and 103, marked in the figure, since this pair of particles has a gap of only 0.024 radii between them. This is not surprising given that we do not include pairwise lubrication corrections in our method; such corrections are included in the reference results to which we are comparing so they produce accurate forces for all particles. In the rigid multiblob method, we resolve the near-field interactions more and more accurately as we increase the number of blobs per sphere, but we cannot accurately resolve

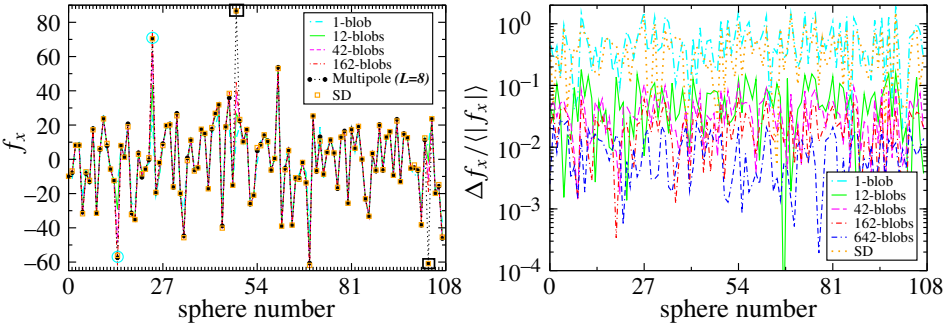


Figure 9. Results from the rigid multiblob method applied to Ladd's benchmark resistance problem (with specified random velocities) for a periodic suspension of 108 spheres. Left: the x-component of the force on each sphere in a random suspension at a low volume fraction $\phi = 0.05$. For comparison, we show the results of Stokesian dynamics (SD) [65] and the HYDROMULTIPOLE code [24] with $L = 8$ moments retained. Two particles that happen to be at a distance closer than 2.02 radii from each other are marked by a black box and a blue circle, and develop unresolved strong lubrication forces between them. Right: the normalized error $|f_x - f_x^{(\mathrm{ref})}|/\langle|f_x^{(\mathrm{ref})}|\rangle$ in the x-component of the force for an FCC lattice at the high volume fraction $\phi = 0.45$. The HYDROMULTIPOLE code with $L = 8$ moments is used as a high-accuracy reference calculation.

the hydrodynamic interactions between pairs of particles with overlapping blobs (see Figure 2).

At the high packing fraction $\phi = 0.45$ there are many pairs of nearly touching particles in a typical random suspension of hard spheres in the absence of (electrostatic) repulsive forces, and there is no hope that the rigid multiblob method can accurately compute the interparticle forces.[9] Therefore, at this density we focus on an FCC lattice configuration, and compare to the multipole-based code with $L = 8$ moments. Here the closest particle distance is 2.36 radii and our method is able to resolve the forces relatively well, especially with more than 12 blobs per sphere; see the right panel in Figure 9. This is perhaps not surprising; however, the more important point we wish to make is that the SD results are now not significantly more accurate than the results obtained from using only a single blob per sphere. The addition of stresslets and pairwise lubrication does not appear to help much in resolving the many-body far-field hydrodynamic multiple scattering in this lattice configuration. Using an icosahedral rigid multiblob already provides an order of magnitude improvement in the typical error over an FTS truncation, and provides an error comparable to keeping $L = 3$ moments in the HYDROMULTIPOLE method (not shown), which is also the minimum number of moments necessary to keep

[9]The results from HYDROMULTIPOLE suggest that even with $L = 15$ moments, which is the maximum that could be afforded with 32 GB of memory since the linear system to be solved is dense and has about $7 \cdot 10^4$ unknowns, convergence is not achieved to sufficiently high accuracy for the random suspension at $\phi = 0.45$.

to capture all long-ranged hydrodynamic interactions, as well as to model active sphere suspensions [119; 50].

6.3. ***Sphere in a slit channel.*** In this section we compute the parallel and perpendicular translational mobilities of a sphere in a slit channel of thickness $19.2R_h$ as a function of the height H of the sphere center above one of the walls. This problem is of relevance to a number of experiments involving spherical colloids confined between two glass microscope slips, and has also been used as a benchmark problem for boundary-integral calculations in [96]. Since the immersed boundary method used here cannot be used for infinite domains, we take a domain of dimensions $(76.8, 19.2, 76.8)R_h$ and apply no-slip conditions on the y boundaries and periodic conditions in the other two directions.

There are no manageable theoretical results accurate for all distances from the wall and all channel dimensions [123]. For the parallel component of the mobility, Faxén obtained exact series expansions for the mobility at the half- and quarter-channel locations, which we use to benchmark our calculations, neglecting the corrections coming from the use of periodic boundary conditions in the directions parallel to the walls. For other positions of the sphere we employ the modified coherent superposition assumption (MCSA) approximation given in (9) of [11], with the mobility relative to a single wall given by the same theoretical lines shown in Figure 5. The rigid multiblob models used in this study have been chosen to give a blob spacing close to $s \approx 2a \approx 3h$, while ensuring that the number of grid cells is integer given the target channel width relative to the effective hydrodynamic radius of the sphere R_h for all of the resolutions studied: a single blob as a minimal model of the sphere [33] (grid size $128 \times 28 \times 128$), 12 blobs (grid size $256 \times 64 \times 256$ and geometric radius $R_g = 2.503h$, giving spacing $s/h = 2.63$), and 42 blobs ($512 \times 128 \times 512$ grid, $R_g = 6.047h$, and $s/h = 3.30$).

We have already confirmed that our results are in agreement with Faxén's theory in the (extrapolated) limit of an infinitely long channel in our prior work [71]. Here we examine the mobility as the sphere moves through the channel, and in particular, as it comes close to the wall. The results of our calculations are shown in Figure 10, and are in good agreement with the approximate MCSA theory for the parallel mobility. Note that the MCSA theory is approximate even far from the wall, as seen from the fact that our results do not match it for the perpendicular mobility even at the center of the channel. The boundary condition handling described in Appendix D of [71] ensures that for a single blob the mobility vanishes at the boundary, i.e., for $H = 0$, rather than for $H = R_h$ as for a true sphere. The more resolved models, however, do show a sharp drop in mobility when the sphere nearly touches the wall. The inset shows that the lubrication interactions are not resolved very close to the wall, just as we observed for a single wall in Section 5.1. Nevertheless, we note that unlike the

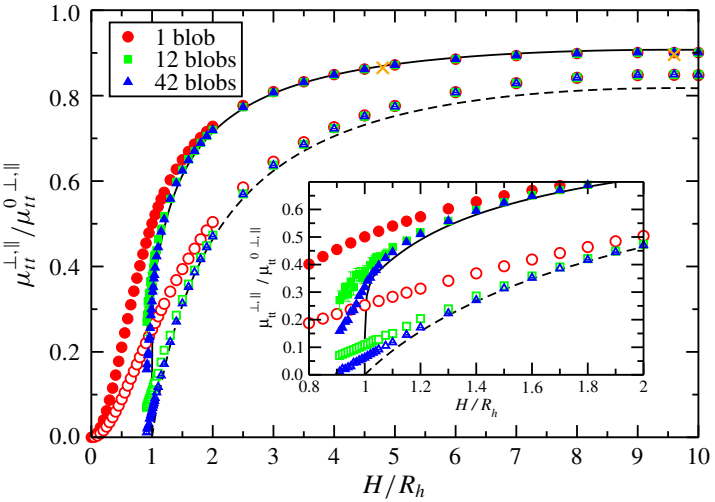


Figure 10. Translational mobility of a sphere in a slit channel of width $19.2R_h$ relative to the bulk, for several resolutions of the rigid multiblobs (see legend), for forces/motion parallel (filled symbols) and perpendicular (empty symbols) to the wall. The numerical results are in good agreement with the exact results of Faxén for distances $H = L/4$ and $H = L/2$ (orange crosses). The inset shows that close to the wall the results for the 12- and 42-blob shells are in reasonable agreement with the approximate MCSA theory (lines).

Rotne–Prager–Blake mobility tensor computed by Swan and Brady [122] and used in Section 5.1, the mobility computed by the grid-based Stokes solver is physically realistic even when the blobs overlap the wall. This has two important implications. First, the rigid multiblob IB method does not run into singularities for $H_{\min} < H < R_h$, where $H_{\min}$ is the distance at which the center of some blob first leaves the physical domain. This is particularly beneficial in Brownian simulations, where stochastic motion can push the sphere to slightly overlap the wall [33; 34]. Furthermore, the immersed boundary results in the inset of Figure 10 are substantially more rotationally invariant as the sphere approaches the wall than the corresponding results in the top two panels of Figure 5; the error bars in the mobility due to discretization artifacts are very small for the immersed boundary method even for $H < R_h$.

6.4. ***Boomerang in a slit channel.*** In this section we study the diffusion of a boomerang colloidal particle in a narrow slit channel, as recently studied experimentally and theoretically [21; 22; 23]. The boomerangs are confined to essentially remain in the plane parallel to the wall by the tight confinement, and thus perform quasi-two-dimensional diffusion. We previously studied the diffusion of such a boomerang colloid sedimented against a single no-slip boundary in [34], using a strong gravitational force to keep the boomerang in quasi-two dimensions. However, the colloids in the actual experiments are almost neutrally buoyant and a slit channel is used to confine them to two dimensions. Here we use our rigid multiblob IB

method to determine the effective two-dimensional diffusion coefficients of a single boomerang in slit confinement.

As discussed in detail in [34], it is, in principle, necessary to perform long Brownian simulations to determine the long-time diffusion tensor $\boldsymbol{D}$ of nonspherical particles. However, if the mean square displacement (MSD) is linear in time, the long- and short-time diffusion coefficients are equal and can be obtained from the Stokes–Einstein relation

$$\boldsymbol{D} = k_B T \langle \boldsymbol{\mathcal{N}} \rangle = k_B T \bar{\boldsymbol{\mu}}, \tag{29}$$

where $\bar{\boldsymbol{\mu}}$ is the average mobility over configurations following the Gibbs–Boltzmann (GB) distribution. Therefore, the diffusion coefficient can be computed by generating samples from the GB distribution of boomerang configurations, and then solving a mobility problem for each configuration and averaging over the samples. For quasi-two-dimensional diffusion the MSD can be made nearly linear by a careful choice of the *tracking point* [34; 21; 22], which is the point whose translation is measured and around which torques are expressed [47]. Chakrabarty et al. [21] have shown that for particles diffusing in two dimensions, the optimal choice of tracking point is the center of hydrodynamics stress (CoH), the location of which can be computed from the bulk mobility tensor [34].

To compare with the experimental results of Chakrabarty et al., we compute the diffusion coefficient of a single boomerang colloidal particle between two walls. In the experiments [21], colloidal particles with boomerang shape diffuse in a channel of width $\sim 2\ \mu$m. The boomerang particles, produced by photolithography, have two arms of length $2.1\ \mu$m, thickness $0.51\ \mu$m, and width $0.55\ \mu$m forming a right angle. In our computations, we use no-slip boundary conditions on the walls of the channel and periodic boundary conditions in the directions parallel to them. We construct two rigid multiblob models of such boomerangs (see [34] for geometric details): a minimally resolved model with 15 blobs (grid size $128 \times 9 \times 128$ and blob spacing $s/h = 1.36$) that is essentially a bent version of the cylinder model shown in the leftmost panel of Figure 1, and a moderately resolved model with 120 blobs (grid size $256 \times 18 \times 256$ and blob spacing $s/h = 2.22$), shown in the rightmost panel of Figure 1.

We assume a hard-core potential between each of the blobs and the walls, and average the mobility over 100 samples generated from the Gibbs–Boltzmann distribution using an accept-reject Monte Carlo procedure [34]. In the experiments, there is likely an additional electrostatic repulsion from the wall; we have checked that adding a short-ranged Yukawa-type repulsion with the walls does not change our results significantly.[10] Following [21], we report the diffusion coefficients computed

[10] In fact, our computations (not shown) indicate that a rather accurate estimate of the average mobility can be computed quickly by simply evaluating the mobility of a boomerang lying exactly on the center plane of the channel.

	Experiments	Ratio = (Experiment/Simulation) 15 blobs	120 blobs
d	1.16 (μm)	1.06	1.06
D_{11}	0.049 (μm^2/s)	0.55	0.47
D_{22}	0.058 (μm^2/s)	0.50	0.46
D_θ	0.044 (rad^2/s)	0.46	0.46

Table 6. Comparison of experimentally measured diffusion coefficients for a boomerang particle in a slit channel to numerical estimates obtained from the rigid multiblob IB method. The tracking point is chosen to be the CoH, which is a point on the boomerang bisector line at a distance d (first row) from the crossing point of the two boomerang arms. We report the translational diffusion coefficients D_{11} and D_{22} in the continuous body frame (CBF) of reference as in [21], averaged over 100 samples from the Gibbs–Boltzmann distribution of particle configurations, for two different resolutions.

using (29) in the continuous body frame (CoB) [21] attached to the colloidal particle, such that the axis X_1 goes along the line that bisects the boomerang angles and the axis X_2 is orthogonal to X_1 (see Figure 1 in [21]). The diffusion coefficients for the boomerang particle are given in Table 6. We see that the computed location of the CoH is in good agreement with experimental estimates. However, both translational and the rotational in-plane diffusion coefficients computed in the simulations are twice as large as those measured experimentally, for both resolutions.

To investigate this large mismatch between simulations and experiments, we explore further the difference between the right-angle boomerangs used in the two distinct experiments [21] (arms of length 2.1 μm and width 0.55 μm) and [23] (arms of length 2.33 μm and width 0.7 μm), both for a reported channel width of 2 μm. The boomerangs in [23] are reported to be about 10% larger than those used in [21]; however, the reported diffusion coefficients are reported to be about 25% larger. This is inconsistent with a purely hydrodynamic model, since the larger particles should have smaller bulk diffusion coefficients and are more confined. Therefore, the larger particles must have a translational diffusion coefficient that is *more* than 1.1 times smaller for translation, and *more* than $1.1^3 \approx 1.33$ times smaller for rotation, if particle size is the only difference between the two experiments. Indeed, in our simulations the translational diffusion coefficient is about 1.25 times smaller for the larger particles, and the rotational one is 1.57 times smaller. This suggests that there are some unreported experimental effects that are not taken into account in the simulations, such as a potentially nonuniform channel thickness or polydispersity in the particles. More careful future investigations are required to understand the origin of the difference between simulations and experiments reported in Table 6.

6.5. ***GMRES convergence.*** In this section we investigate the performance of the preconditioner described in Section 3.3 and determine optimal values for $N_s^{(1)}$

and $N_s^{(2)}$, the number of iterations in the first and second approximate Stokes solves in the preconditioner. As a Krylov solver, here we use the restarted right-preconditioned GMRES method, but we have also observed good performance with the short-term recurrence BiCGStab method, which typically requires a few more iterations than GMRES but has smaller memory requirements. It is important to emphasize that the exact number of iterations depends strongly on the geometry of the rigid multiblobs, notably, the spacing between the blobs. The performance depends even more strongly on the efficacy of the geometric multigrid preconditioner for the Poisson equation, which in the implementation used here is strongly degraded for grids that have a nearly prime number of cells in each dimension, and for grids of large aspect ratios (even if all directions are powers of 2). Our focus here is on investigating the trends in the number of GMRES iterations with the various parameters in the preconditioner and the system size.

The computational cost of the solver is dominated by the application of the full preconditioner, whose complexity depends on nontrivial ways on its different steps and on the parameters of the simulations. However, in most cases the cost is dominated by the multigrid cycles for the Poisson equation, and each application of the projection preconditioner for the Stokes equation (28) involves $d+1=4$ scalar V-cycles. Here we use preconditioned Richardson iteration as an iterative solver for the (unconstrained) Stokes equations.[11] Therefore, the total number of scalar multigrid cycles per GMRES iteration is $4(N_s^{(1)}+N_s^{(2)})$ and we can use $N_s^{(1)}+N_s^{(2)}$ as a proxy for the computational cost. It should be noted, however, that this is only an approximation and in practice it may be better to allow a small increase on the total number of Stokes preconditioner applications if it reduces significantly the number of outer GMRES iterations.

We study the solver convergence for a random bidisperse suspensions of hard spheres with aspect ratio $R_{h,1}/R_{h,2}=1/2$ at different concentrations and system sizes and geometries. The parameters of the rigid multiblob models are identical to those reported in Section 6.3. We investigate a suspension at a moderate volume fraction $\phi=0.15$ ($\phi_1=\phi_2=0.075$ for the two components), as well as a suspension at a high volume fraction $\phi=0.45$ ($\phi_1=\phi_2=0.225$). We investigate three different types of boundary conditions, a *periodic* suspension in a cubic domain, a suspension in a *slit channel* with periodic boundaries in the directions parallel to the wall, and a *square channel* with periodic boundaries in the direction of the channel axis. The configurations of hard spheres were generated using a Monte Carlo algorithm with hard core exclusion radius equal to the effective hydrodynamic radius of the spheres.

[11] Richardson iteration is not effective as a stand-alone Stokes solver, but as already explained, it is more efficient in this constrained context because we only need a rather approximate Stokes solver for the unconstrained fluid equations.

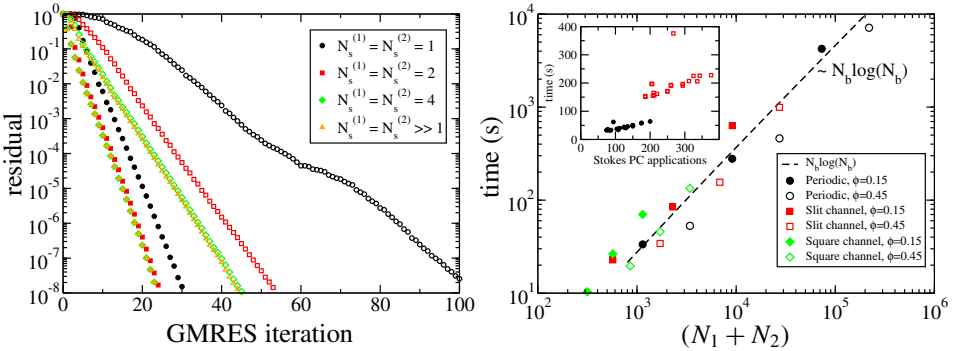


Figure 11. Convergence of GMRES with restart frequency of 60 iterations for a bidisperse suspension of spheres in a cubic periodic domain (filled symbols, grid size 128^3, and $N_s = 1014 + 127 = 1141$ spheres), and in a slit channel of dimensions $4L \times L \times 4L$ (open symbols, grid size $256 \times 64 \times 256$, and $N_s = 6083 + 760 = 6843$ spheres). All spheres are subject to random forces, torques and slips. Left: normalized residual versus number of iterations of the outer solver for different numbers of iterations in the first ($N_s^{(1)}$) and second ($N_s^{(2)}$) Stokes subsolves inside the preconditioner. For comparison, we also solve the Stokes subproblems to high accuracy using an inner iteration of GMRES, marked $N_s^{(1)} = N_s^{(2)} \gg 1$ in the legend. Right: total computing time to solve the mobility problem as a function of the number of spheres $N_s = N_1 + N_2$ using an implementation based on the IBAMR library and 8 cores of an Intel Xeon (E5-2665 2.4 GHz) processor. The inset shows the time to solve the linear system versus the total number of Stokes preconditioner applications $(N_s^{(1)} + N_s^{(2)})\, N_{\text{iter}}$.

In the left panel of Figure 11 we show the relative residual versus the number of iterations of the outer solver for different values of $N_s^{(1)}$ and $N_s^{(2)}$. Because of all of the approximations in the analytical blob-blob mobility matrix, the convergence does not improve with increasing $N_s^{(1,2)}$ beyond some point. Therefore, it is not necessary to perform nearly exact Stokes solves (e.g., complete FFTs in periodic domains) inside the preconditioner; a few (spectrally equivalent) cycles of multigrid are sufficient. In the inset in the right panel of Figure 11 we show that the total number of applications of the Stokes preconditioner $N_s = (N_s^{(1)} + N_s^{(2)})N_{\text{iter}}$ is a reasonable proxy for the computational time, where N_{iter} is the number of GMRES iterations.[12]

Table 7 shows the number of GMRES iterations[13] required to reduce the residual by a factor of 10^8 for a variety of system sizes, keeping $N_s^{(1)} = 2$ and $N_s^{(2)} = 1$. As seen in the table, the convergence for periodic domains, just as for the methods based on Green's functions studied in Sections 4.4 and 5.4, is independent of the system size and only depends weakly on the packing density. The largest system

[12] The actual cost has the form $aN_s + bN_{\text{iter}}$ where b grows with the number of blobs; therefore, two outliers are observed for $N_s^{(1)} = 1$ and $N_s^{(2)} = 0$ since these require a large number of GMRES iterations to converge.

[13] GMRES is restarted every 60 iterations except for the largest system (512^3 fluid cells and about $2 \cdot 10^5$ particles) which uses a restart frequency of 20 to reduce memory requirements.

ϕ	Periodic Cells	N_1	N_2	N_G
0.15	$128 \times 128 \times 128$	1 014	127	28
	$256 \times 256 \times 256$	8 111	1 014	29
	$512 \times 512 \times 512$	64 885	8 111	29
0.45	$128 \times 128 \times 128$	3 041	380	42
	$256 \times 256 \times 256$	24 332	3 041	43
	$512 \times 512 \times 512$	194 656	24 332	44

ϕ	Square channel Cells	N_1	N_2	N_G	Slit channel cells	N_1	N_2	N_G
0.15	$128 \times 64 \times 64$	283	32	37	$128 \times 64 \times 128$	507	63	34
	$256 \times 64 \times 64$	507	63	47	$256 \times 64 \times 256$	2 028	253	42
	$512 \times 64 \times 64$	1 014	127	65	$512 \times 64 \times 512$	8 111	1 014	63
0.45	$128 \times 64 \times 64$	760	95	58	$128 \times 64 \times 128$	1 521	190	52
	$256 \times 64 \times 64$	1 521	190	76	$256 \times 64 \times 256$	6 083	760	62
	$512 \times 64 \times 64$	3 041	380	119	$512 \times 64 \times 512$	24 332	3 041	96

Table 7. GMRES convergence results for a bidisperse suspension in periodic and confined domains. A random configuration of N_1 hard spheres of radius $R_h = 1$ (12 blobs) and N_2 hard spheres of radius $R_h = 2$ (42 blobs) is generated, and random forces, torques, and slips are applied to all of the particles. We report the number of GMRES iterations N_G needed to reduce the residual by a factor of 10^8 for the mobility problem for a variety of system sizes and boundary conditions.

has a grid of 512^3 cells and almost 3.4 million blobs on $2.2 \cdot 10^5$ spherical shells packed to a rather high density $\phi = 0.45$, yet the GMRES iteration converges after only 44 iterations. For confined systems the solver requires more iterations, as expected because the boundary conditions are not taken into account in either the Stokes solver preconditioner or the block-diagonal mobility approximation. At first sight, it may appear that the number of iterations grows with the system size for nonperiodic domains. This increase, however, comes not because of the increase in system size but rather because the aspect ratio of the domain grows and the multigrid algorithm used in our implementation becomes less effective. This can be confirmed by noting that the number of iterations grows very weakly with system size if we keep the domain aspect ratio fixed; for a square channel and $\phi = 0.45$ we require 58 iterations for a grid with $128 \times 64 \times 64$ cells, 60 iterations for $256 \times 128 \times 128$ cells, and 65 iterations for $512 \times 256 \times 256$ cells (compared to 119 for $512 \times 64 \times 64$ cells). This demonstrates that our preconditioner robustly handles large system sizes even in the presence of physical boundaries.

We believe the performance of the solver for high-aspect-ratio domains can be greatly improved with a new multigrid implementation capable of dealing with highly noncubic domains at the coarsest levels of the multigrid hierarchy. The right panel of Figure 11 shows the total computing time as a function of system size and demonstrates the near-linear scaling of the method at fixed computing power, at least for cubic periodic domains, for which the multigrid implementation used in the IBAMR library is nearly optimal.

6.6. ***Sedimentation velocity in a binary sphere suspension.*** In our last test we use our rigid multiblob IB method to compute the mean and variance of the instantaneous sedimentation velocity in a random binary suspension of hard spheres, as done using Stokesian dynamics (SD) by Wang and Brady [131; 132]. The binary suspension has two components, $\alpha = 1$ and $\alpha = 2$, with equal volume fractions $\phi_1 = \phi_2 = \phi/2$ and size ratio $R_2/R_1 = 2$. The two types of particles are assumed to be much denser than the solvent and to have the same density, so that the ratio of the gravitational forces is set to $F_2/F_1 = 8$. Here we average the sedimentation velocity statistics over an ensemble of sphere packings that are sampled from the equilibrium distribution in the absence of gravity. To generate configurations of spheres, we use the Lubachevsky–Stillinger packing algorithm [37; 38] to create an initial packing of spheres, and then use equilibrium event-driven hard-sphere molecular dynamics to equilibrate the packings. We then apply gravitational forces on all spheres and solve the mobility problem to compute the instantaneous sedimentation velocities $U_{s,\alpha}$ for each species. As described in more detail in [71], the total gravitational force on the spheres must be balanced by an equal and opposite uniform force density in the fluid because of the use of periodic boundary conditions.

The rigid multiblob models used in this study have either 12 blobs ($R_g = 0.6643$, $R_h \approx 1$, and $s/h = 2.052$) for the smaller species and 42 blobs for the larger species ($R_g = 1.7714$, $R_h \approx 2$, and $s/h = 2.843$) or, for improved resolution, 42 blobs for the smaller species ($R_g = 0.8692$, $R_h \approx 1$, and $s/h = 2.553$) and 162 blobs for the larger species ($R_g = 1.8935$, $R_h \approx 2$, and $s/h = 2.808$). To correct for finite system size effects, for each volume fraction ϕ we run simulations for 3 grid resolutions; specifically, we use grids of sizes 64^3, 128^3, and 256^3 cells for the smaller resolution (12–42 blobs per sphere), and 128^3, 256^3, and 512^3 for the higher resolution (42–162 blobs). The average sedimentation velocity was extrapolated to the infinite system size limit by assuming that the finite-size corrections scale as $N^{-1/3}$, where N is the total number of particles, instead of assuming a specific analytical form for the corrections [131; 85]. The largest example in our simulations is for $\phi = \frac{1}{2}$ with a 512^3 grid, for $N_1 = 51\,200$ and $N_2 = 6\,400$ spheres, corresponding to a total of about 10 million Lagrangian (i.e., blob/body) degrees of freedom (DoFs), and about half a billion Eulerian (i.e., fluid) DoFs.

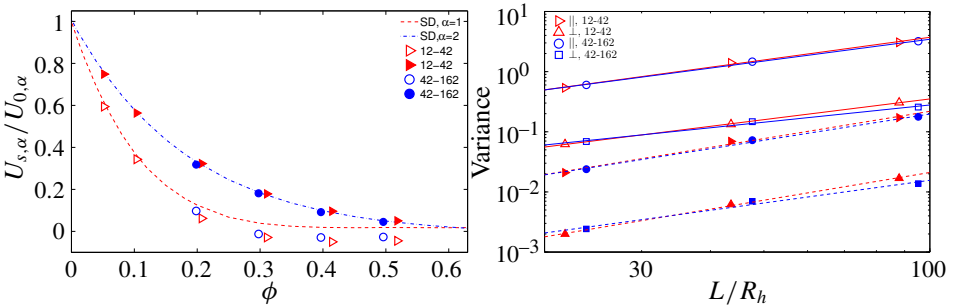


Figure 12. Instantaneous sedimentation rates of the two species, $\alpha = 1$ (empty symbols) and $\alpha = 2$ (filled symbols), in a binary suspension of hard spheres, for two different resolutions (see legend). Left: average vertical sedimentation velocity normalized by $U_{0,\alpha} = F_\alpha/(6\pi\eta R_\alpha)$ as a function of the total volume fraction ϕ. The data from recent Stokesian dynamics simulations [131] are shown as lines. Right: normalized variance $\Delta U_\alpha^2 = \langle \delta U_{s,\alpha}^2 \rangle / U_{0,\alpha}^2$ of the sedimentation velocity parallel and perpendicular to gravity for $\phi \approx 0.2$. Linear fits to the data are shown as dashed lines.

The left panel of Figure 12 compares our results for the mean sedimentation velocity of the different species with results obtained using traditional (i.e., non-accelerated) SD without pairwise lubrication corrections [131]. It is well-known that a standard FTS truncation is not particularly accurate for sedimentation because of the importance of a nontrivial mean quadrupole [15]. Therefore, the SD simulations include a mean-field estimate of the quadrupole contribution; see (2.29) in [16]. Such a correction is not included in the accelerated SD method developed in [132], and this leads to a strong overestimation of the sedimentation velocity at larger densities and even a reversal of the trend toward increasing sedimentation rate [132]. Our results show a consistently decreasing sedimentation rate with increasing density, and are in good agreement between the two resolutions, except that the agreement is only qualitative at the higher densities for the smaller spheres (thus indicating a lack of convergence in our numerical results). Our results are consistent with the SD results for the larger particles over the range of densities studied here. However, for the smaller particles we find a smaller sedimentation rate and even a negative rate, which arises due the strong backflow created around the larger particles. As discussed in Section 6.2, lubrication forces can be very important at densities as large as $\phi = \frac{1}{2}$, although they are often assumed to play little role in sedimentation due to lack of relative motion among the particles, and are therefore not included in the SD simulations. Nevertheless, it may be that lubrication forces play a role for dense binary suspensions due to the relative motion of the small spheres around the large spheres. We therefore believe that the binary sedimentation problem should be revisited by more accurate methods or experiments.

The right panel of Figure 12 shows the normalized variance of the instantaneous sedimentation velocities for the two species at $\phi = 0.2$ as a function of system size.

Consistent with theory and simulation for random suspensions of monodisperse suspensions [85], we find that the variance grows linearly with system size, consistently between the two resolutions. This unphysical growth has been the subject of a long-standing controversy in the literature, which cannot be resolved by our static (i.e., instantaneous) computations. Namely, it has been noted that the structure of the suspension changes during sedimentation [83], although not enough to suppress the variance growth in existing lattice Boltzmann (LB) simulations [85]. More recent LB studies have suggested that boundaries, polydispersity, and stratification all play roles in the sedimentation of a realistic suspension [103].

7. Conclusions

In this paper we described a numerical method for simulating non-Brownian Stokesian suspensions of passive and active rigid particles of essentially arbitrary shape in either unconfined, partially confined, or fully confined geometries. Following a number of prior works, we discretized rigid bodies using a collection of minimally resolved spherical blobs to move as a rigid body. A key contribution of our work was the development of preconditioned iterative solvers for the potentially large linear system of equations for the unknown Lagrange multipliers $\boldsymbol{\lambda}$ and rigid-body motions $\boldsymbol{U}$. We demonstrated that an effective and scalable approach is to solve the saddle-point problem for both $\boldsymbol{\lambda}$ and $\boldsymbol{U}$ using a block-diagonal preconditioner that ignores hydrodynamic interactions of distinct bodies, or even interactions between the bodies and the boundary.

The hydrodynamic interactions between the blobs are captured using the Rotne–Prager–Yamakawa (RPY) tensor tailored to the specific geometry (boundary conditions). For unbounded domains, we used a fast multipole method to compute the product of the blob-blob mobility $\mathcal{M}$ and a force vector. For a single no-slip boundary, we used a GPU to directly sum the Rotne–Prager–Blake tensor over all pairs of blobs; FMM methods for half-space Stokes flow have recently been developed [51] and could be used to scale these computations to millions of blobs. We showed empirically that the number of GMRES iterations required to solve for $\boldsymbol{\lambda}$ and $\boldsymbol{U}$ is bounded independent of the number of bodies, and grows only weakly with increasing packing density. This paves the way for the development of linear-scaling methods for solving the mobility problem in moderately dense suspensions of hundreds of thousands of particles. At the same time, we find that solving the resistance problem is substantially more difficult since the number of iterations grows approximately linearly with the linear dimensions of the system.

For more complex boundary conditions such as fully confined domains, there is no simple analytical approximation to the RPY tensor [99]. While it is possible to construct fast methods for computing the product $\mathcal{M}\boldsymbol{\lambda}$ in *specific* geometries,

e.g., using Ewald summation for periodic domains [126], this requires knowing the Green's function analytically, and more importantly, requires a new method and code for each specific combination of boundary conditions. As an alternative, in this work we developed a rigid multiblob method for periodic suspensions or suspensions confined in slit and square channels that uses a grid-based Stokes solver to compute the action of the (regularized) Green's function "on the fly" [33; 58; 135]. Specifically, we extended a recently developed rigid-body immersed boundary method [71] to suspensions of freely moving passive or active rigid particles at zero Reynolds number. We demonstrate that GMRES applied to the coupled fluid plus rigid body equations converges in a bounded number of iterations independent of the system size, with a weak growth of the number of iterations with the packing density, and a moderate growth with increased confinement. Unlike in methods based on Green's functions, each Krylov iteration in our approach only requires a few cycles of a geometric multigrid solver for the Poisson equation, and an application of the block-diagonal preconditioner for the blob-blob mobility.

We used our methods to compute the mobility of a cylinder near a no-slip boundary and found good agreement with experimental measurements. We also demonstrated that a pair of active pusher tripartite nanorods sedimented near a boundary form dimers that rotate in a direction consistent with recent experimental measurements [29]. Our numerical results for the effective planar diffusion coefficient of a boomerang colloid confined to a narrow slit channel were not in agreement with recent experimental measurements [21; 23] by a factor of 2. In the future we will carry out more careful and systematic quantitative comparisons between simulations and experiments for confined passive and active colloids.

It is worthwhile to point out some specific differences between our approach and existing methods. We focus in this comparison on methods based on Green's functions. For a confined domain, our Green's-function-free method described in Section 3 is quite different from most existing methods. The equations (10) appear, perhaps in somewhat modified form, in a number of works [104; 48; 46; 62; 25; 80; 35]. The key distinguishing feature of our work is the use of iterative methods as a way to scale these computations to suspensions of thousands of bodies. While preconditioned iterative solvers have been used in the recent work of Swan and Wang [126], we believe the preconditioned saddle-point approach developed here is notably superior both in efficiency *and* simplicity.

The rigid multiblob approach is quite similar to the method of regularized Stokeslets developed by Cortez et al. [28; 27; 2; 88]. This method is usually presented as a regularized first-kind boundary-integral formulation [28] for solving (A-5). The method has been made more accurate by using higher-order quadratures [120; 101], and has very recently been generalized to second-kind formulations that account for active slip [101]. However, these works do not consider preconditioners

and existing regularized Stokeslet methods do not scale well to many-body suspensions. We note that a first-kind formulation preconditioned by a block-diagonal preconditioner as we use in this work is spectrally equivalent to a second-kind formulation for well-separated bodies.[14] In [101] double-layer terms (i.e., second-kind boundary integrals) are included to account for the active slip. As we argue in Appendix A.1, this is not necessary if one is not interested in surface tractions, and therefore, we prefer our simpler regularized first-kind formulation. Another key difference between our approach and the method of regularized Stokeslets is that the mobility used in regularized Stokeslets methods is different from the RPY tensor; most importantly, it is *not* symmetric. Notably, Cortez et al. apply the regularization on the forces (sources) but not also on the velocities (targets), which approximately corresponds to omitting $(\boldsymbol{I} + a^2/6\,\nabla_{\boldsymbol{r}}^2)$ in (7). Using a nonsymmetric blob-blob mobility is not physical; for example, incorporating thermal fluctuations becomes impossible since this requires the square root of the mobility.

Our work is very closely related to that of Swan, Brady, et al. [122]. The following are key differences. First, we use only the RPY form of the mobility matrix; that is, we only have a force (monopole) degree of freedom at each blob, as in more recent work by Swan and Wang [126]. This can be seen as a direct but regularized discretization of (A-5), where the unknown is the surface "traction". By contrast, Swan et al. use Stokesian dynamics (SD) to represent the blobs as "spheres", more precisely, to associate with each blob a force, torque, and stresslet (FTS);[15] more multipoles have been included in other works based on multipole expansions [25; 106; 95]. This makes the number of degrees of freedom (DoF) per blob at least $3 + 3 + 5 = 11$ in three dimensions, instead of just 3 as in our formulation. In recent work [35], rotational degrees of freedom (angular velocity and constraint torques) have been added to the blobs without including stresslets (i.e., an FT truncation), which doubles the number of DoFs per blob relative to the approach we follow (6 DoFs instead of 3). Our investigations have shown this to lead to an insufficient improvement in accuracy to justify the doubling of the number of DoFs. For active suspensions, in the formulation of [106; 119; 118], active slip is imposed on the surface of the beads composing the rigid body; i.e., each bead is active *individually*. By contrast, our blobs do not really have a well-defined surface, and in our formulation active slip is imposed on the surface of the body and not on blobs individually, consistent with a discretization of (A-5). Our approach *only* requires a way to compute the (action of the) RPY blob-blob mobility matrix; it is therefore much simpler to use in practice and it adapts to different boundary conditions. As we explained in Section 3, the RPY tensor can be approximated

[14] We thank Leslie Greengard for sharing this observation with us.

[15] Note that a degenerate quadrupole correction corresponding to the Faxén operators in (7) is also included in the RPY tensor even for "monopoles".

using grid-based solvers quite straightforwardly using immersed boundary methods, but going to higher orders requires additional differentiability (smoothness) [92] than afforded by simple immersed boundary methods.

Another key difference between the rigid multiblob method and traditional Stokesian dynamics is that we do *not* include lubrication (near-field) corrections in addition to the far-field RPY mobility. If it is necessary to resolve near-field interactions between particles, for example, to study the rheology of concentrated suspensions, one can increase the resolution by using more blobs per rigid body. For sufficiently dense suspensions, very close contacts become numerous and in practice lubrication forces need to be included as a correction to the FTS expansion. We choose, however, not to include *uncontrolled* pairwise lubrication approximations for several reasons. First, we believe that it is important to control the approximations so that accuracy can be confirmed by comparing different levels of resolution. Second, it is difficult to generalize pairwise lubrication corrections to dense suspensions of rigid particles of *arbitrary* shape.

We carefully studied the accuracy of the rigid multiblob approach on a variety of standard problems for spherical particles. We demonstrated that, once the effective hydrodynamic radius of the rigid multiblobs is matched to the target sphere radius, even a 12-blob (icosahedral) model of a sphere [129] provides a substantial improvement over the widely used force-torque-stresslet (FTS) truncation of the multipole hierarchy, especially near boundaries. However, we note that the rigid multiblob models are not rotationally invariant and this leads to notable discretization artifacts as blobs on distinct bodies begin to overlap. Furthermore, our method does not include pairwise lubrication corrections for nearby pairs of spheres (for reasons discussed in the body of the paper), and can therefore only accurately resolve the hydrodynamic interactions between objects if the blobs on the two bodies do not overlap each other. It remains a grand challenge for future work to construct a *scalable* method that applies to particles of complex shape with complex boundary conditions *and* resolves lubrication interactions among nearly touching particles with *controllable* accuracy.

There are a number of possible extensions of the computational method described here. An important direction of work is to compute a tractable formulation of the RPY–Blake tensor for a single no-slip boundary that ensures an SPD mobility matrix even when blobs overlap the wall, which is important for the inclusion of thermal fluctuations (Brownian motion). While a general SPD formulation of RPY in confined domains has been developed in [99], that formulation does not apply a regularization when the blobs overlap the wall.[16] Such a regularization is important physically; in particular, we believe it is crucial that the velocity of a blob go to zero smoothly as its position approaches the boundary. This prevents

[16] In fact, the overlapping correction derived in [99] is *independent* of the boundary conditions.

unphysical motion of blobs along the no-slip boundary, or even worse, blobs leaving the domain. Observe that the alternative formulation of the blob-blob mobility (21), together with the modification near no-slip boundaries first proposed by Yeo and Maxey [134] and generalized to other boundary conditions in Appendix D of [71], is SPD for all configurations *and* vanishes as a blob approaches a boundary. If the integral in (21) can be performed analytically for some choice of the kernel δ_a, this would give a simple formula for a Rotne–Prager–Yamakawa–Blake tensor that can be used in practical simulations. Another approach to constructing a regularization is to use the simple image construction proposed recently by Gimbutas et al. [51] and combine with the free-space RPY tensor.

The rigid multiblob formulation can be seen as a low-order regularized quadrature rule for the first-kind integral equation (A-5). It is natural to consider using higher-order quadrature rules. This has been done in the context of the method of regularized Stokeslets in [120; 101], and has been done in the context of immersed boundary methods in [55]. Specifically, Griffith and Luo have proposed an alternative IB approach that models the deformations and stresses of an immersed elastic body using a finite-element (FE) representation [55]. In their IB/FE approach, the degrees of freedom associated with $\boldsymbol{\lambda}$ are represented in a finite-element basis set, and the interaction between the fluid grid and body mesh is handled by placing IB markers at the numerical quadrature points of the FE mesh. When such an approach is generalized to rigid bodies, it simply amounts to *filtering* the mobility operator (26):

$$\mathcal{M}_{\mathrm{FE}} = \boldsymbol{\Psi}(\mathcal{J}\mathcal{L}^{-1}\mathcal{S})\boldsymbol{\Psi}^T = \boldsymbol{\Psi}\mathcal{M}\boldsymbol{\Psi}^T,$$

where $\boldsymbol{\Psi}$ is a matrix that contains quadrature weights as well as geometric information about the FE mesh. Future work should explore whether this approach provides a significant improvement in accuracy or efficiency over the simple rigid multiblob approach presented here, and compare this to the methods described in [120; 101].

In the method used here, we used a regular (staggered) grid, and therefore, no-slip boundary conditions can only be imposed on the boundaries of a rectangular prism. Domains of complex shapes, such as (patterned) microfluidic channels, can be handled in two ways. The first way is to construct the boundaries out of rigidly fixed blobs [135]. While this is flexible and straightforward, it requires solving a combined mobility-resistance problem that our investigations suggest cannot be solved scalably using existing methodologies. An alternative and promising approach is to use an FEM method to solve the Stokes equations on a boundary-fitted unstructured tetrahedral grid [109], and combine this with the rigid-body immersed boundary ideas presented here. Even if a rectangular grid is appropriate, our regular-grid method requires very large grids for very low densities or inhomogeneous suspensions, such as, for example, a suspension of particles sedimented near a bottom wall in a slit channel where the top wall needs to be taken into account as

well. A substantial challenge for future work is to develop a stable discretization of the steady Stokes equations on an adaptively refined (e.g., block-structured) staggered grid; this has been accomplished for unsteady flow [54] but the steady Stokes equations pose several notable challenges.

We believe that a number of the preconditioning ideas developed in this work can also be applied to other related methods, such as methods based on boundary-integral formulations. Some of these methods can provide a notable improvement in accuracy over the low-accuracy rigid multiblob method, and with a suitable preconditioner they can potentially be scaled to suspensions of tens of thousands of particles. For certain simple confined geometries, such as periodic boundaries or semi-infinite slit channels, it is possible to develop fast methods for applying the RPY and related tensors based on FMM or FFT methods. This may be preferable to the immersed boundary approach followed here, which requires a dense grid of spacing *smaller* than the hydrodynamic radius of the blobs a. By contrast, the spectral Ewald method [90] completely decouples the spacing of the FFT grid from a, while controlling the accuracy. We believe that is important for the community working on Stokes suspensions to develop benchmark problems and compare different methods in terms of both accuracy and efficiency, to identify which methods are most appropriate under which conditions and accuracy requirements.

To account for thermal fluctuations (Brownian motion), one adds a fluctuating component $(k_B T \eta)^{1/2}(\boldsymbol{\mathcal{Z}}(\boldsymbol{r},t) + \boldsymbol{\mathcal{Z}}^T(\boldsymbol{r},t))$ to the fluid stress $\boldsymbol{\sigma}$ in (A-1) [33; 73; 4; 31] in the spirit of fluctuating hydrodynamics [61; 114; 6], where $\boldsymbol{\mathcal{Z}}(\boldsymbol{r},t)$ denotes a white-noise random Gaussian tensor field with uncorrelated components. This adds a fluctuating component to the rigid-body velocity and leads to the overdamped Langevin equation

$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} d\boldsymbol{q}/dt \\ d\boldsymbol{\theta}/dt \end{bmatrix} = \boldsymbol{\mathcal{N}} \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{\tau} \end{bmatrix} - \breve{\boldsymbol{\mathcal{N}}}\, \breve{\boldsymbol{u}} + (2k_B T \boldsymbol{\mathcal{N}})^{1/2} \diamond \boldsymbol{\mathcal{W}}(t), \tag{30}$$

where $\boldsymbol{\mathcal{W}}(t)$ denotes a collection of independent white-noise scalar processes and $\diamond$ denotes a suitable (kinetic) stochastic product [64; 34]. In this work we did not consider the generation of the fluctuating component of the rigid-body motion $(2k_B T \boldsymbol{\mathcal{N}})^{1/2}\boldsymbol{W}$, where $\boldsymbol{W}$ is a collection of standard normal variates. This is an important missing component for suspensions in an unbounded or half-space domain. For the immersed boundary method described in Section 3, computing the random motion of rigid multiblobs is straightforward and can be accomplished at essentially *no additional cost* by simply including the stochastic stress on the right-hand side in the fluid equations. The difficulty, which we will address in future work, is to develop a temporal integration scheme for (30) that correctly accounts for the stochastic drift terms without incurring significant additional costs compared to non-Brownian suspensions [33; 31; 34].

Acknowledgments

We have benefited greatly from extended discussions with Leslie Greengard, James Swan, Anthony Ladd, Michael Shelley, Megan Davies Wykes, Anna-Karin Tornberg, and Ronojoy Adhikari. We thank Shidong Jiang for sharing with us his RPY FMM code, Manas Rachh and Leslie Greengard for sharing with us the latest plane-wave FMMLIB3D codes, James Swan for sending us Mathematica code to evaluate pairwise sphere mobilities to high accuracy, Anthony Ladd for sharing with us his numerical results on random sphere suspensions, Maciej Długosz for sharing with us results from his code, and John Brady for sharing with us the results of SD for binary sphere suspensions. A. Donev and B. Delmotte were supported in part by the National Science Foundation under award DMS-1418706. A. Donev and F. Balboa Usabiaga were supported in part by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under award number DE-SC0008271. B. Delmotte was supported partially by the Materials Research Science and Engineering Center (MRSEC) program of the National Science Foundation under award number DMR-1420073. B. E. Griffith and A. P. S. Bhalla were supported in part by the National Science Foundation under awards DMS-1460368, ACI-1460334, and ACI-1450327. We thank the NVIDIA Academic Partnership program for providing GPU hardware for performing some of the simulations reported here.

Appendix A: Continuum formulation

The basic problem we consider is the motion of a number of rigid bodies suspended in a Stokesian fluid. For simplicity, consider a single body Ω rotating with angular velocity $\boldsymbol{\omega}$ around a *tracking point* (origin) that is translating with linear velocity $\boldsymbol{u}$, under the combined influence of an external force $\boldsymbol{f}$ and torque $\boldsymbol{\tau}$; the generalization to many bodies is straightforward. Without loss of generality let us assume that the fixed (lab) and body coordinate frames are identical at the point in time under consideration. Outside the body we have the steady Stokes equations for the fluid velocity $\boldsymbol{v}(\boldsymbol{r})$ and the pressure $\pi(\boldsymbol{r})$,

$$\begin{aligned} -\nabla\cdot\boldsymbol{\sigma} &= \nabla\pi - \eta\nabla^2\boldsymbol{v} = 0, \\ \nabla\cdot\boldsymbol{v} &= 0, \end{aligned} \tag{A-1}$$

along with some suitable boundary conditions at infinity or the boundary of a domain $\mathcal{D}\supset\Omega$. The *no-slip* boundary condition on the surface of the body is

$$\boldsymbol{v}(\boldsymbol{q}) = \boldsymbol{u} + \boldsymbol{\omega}\times\boldsymbol{q} + \breve{\boldsymbol{u}}(\boldsymbol{q}) \quad \text{for all } \boldsymbol{q}\in\partial\Omega, \tag{A-2}$$

where $\breve{\boldsymbol{u}}$ is a specified *apparent slip* velocity due to active boundary layers on the surface of the rigid body. Here $\boldsymbol{u}$ and $\boldsymbol{\omega}$ are Lagrange multipliers for the force and

torque balance conditions

$$\int_{\partial\Omega} \boldsymbol{\lambda}(\boldsymbol{q})\, d\boldsymbol{q} = \boldsymbol{f}, \qquad \int_{\partial\Omega} [\boldsymbol{q} \times \boldsymbol{\lambda}(\boldsymbol{q})]\, d\boldsymbol{q} = \boldsymbol{\tau}, \tag{A-3}$$

where $\boldsymbol{\lambda}(\boldsymbol{q})$ is the normal component of the stress on the outside of the surface of the body, i.e., the traction

$$\boldsymbol{\lambda}(\boldsymbol{q}) = \boldsymbol{\sigma} \cdot \boldsymbol{n}(\boldsymbol{q}),$$

where $\boldsymbol{n}$ is the surface normal and the fluid stress is

$$\boldsymbol{\sigma} = -\pi \boldsymbol{I} + \eta(\nabla \boldsymbol{v} + \nabla^T \boldsymbol{v}). \tag{A-4}$$

The solution of the above system of equations is, by linearity, an affine mapping of the form (1).

A.1. ***Boundary integral reformulation.*** Observe that in the Stokes regime, the details of what happens inside the body do not actually matter for the motion of the body and its hydrodynamic interactions with other bodies or boundaries. For instance, a fluid-filled "bacterium" with a rigid membrane and a solid particle of the same shape will move identically for the same surface slip and total force and torque. Similarly, to an outside observer, a bacterium covered with a layer of cilia on the outside will be indistinguishable from a bacterium that also has a layer of cilia on the inside of its membrane. Therefore, it is possible to extend the fluid equation (A-1) to the whole domain and pretend that there is fluid inside the body moving with a velocity that is continuous across the boundary of the body. For a strictly rigid body motion on the surface, the fluid inside will move as a rigid body and thus be free of strain [28]. If there is slip on the surface, when we extend the flow inside we are assuming that the velocity is continuous at the boundary so that the same slip is present on the inside of the body surface. This will drive *internal* active flows inside the body in addition to the *external* active flow outside. Once we extend the fluid equation everywhere we can write down an equivalent *first-kind* boundary-integral equation for Stokes flow [111; 28]

$$\boldsymbol{v}(\boldsymbol{q}) = \boldsymbol{u} + \boldsymbol{\omega} \times \boldsymbol{q} + \breve{\boldsymbol{u}}(\boldsymbol{q}) = \eta^{-1} \int_{\partial\Omega} \mathbb{G}(\boldsymbol{q}, \boldsymbol{q}')\tilde{\boldsymbol{\lambda}}(\boldsymbol{q}')\, d\boldsymbol{q}' \quad \text{for all } \boldsymbol{q} \in \partial\Omega, \tag{A-5}$$

which along with the force and torque balance condition (A-3) defines a linear system of equations to be solved for the single-layer potential $\tilde{\boldsymbol{\lambda}}(\boldsymbol{q})$ and the velocities $\boldsymbol{u}$ and ω. Here $\mathbb{G}(\boldsymbol{q}, \boldsymbol{q}')$ is the Green's function for steady Stokes flow with unit viscosity and with the specified boundary conditions on the domain boundary $\partial\mathcal{D}$.

In this work we will require that the total volume of fluid is preserved by the slip, i.e., there is no source or sink for the flow inside the particle (as would be the

case for swelling bodies):

$$\int_{\partial\Omega} \breve{\boldsymbol{u}}(\boldsymbol{q}) \cdot \boldsymbol{n}(\boldsymbol{q})\, d\boldsymbol{q} = 0, \tag{A-6}$$

which is always true for tangential slip. This condition is required to be able to extend the flow inside the body and still keep it incompressible everywhere in the domain. This condition is related to a known issue with first-kind boundary-integral formulations having a nontrivial null space or, equivalently, the single-layer operator having an incomplete range [111]. Removing the restriction (A-6) requires switching to a second-kind or a mixed first-second-kind integral equation [101; 119].

The single-layer potential $\tilde{\boldsymbol{\lambda}} \equiv \boldsymbol{\lambda} = \boldsymbol{\sigma} \cdot \boldsymbol{n}$ if there is no slip, i.e., if $\breve{\boldsymbol{u}} = 0$, which is a property that relies closely on the fact that the specified velocity on the surface of the body is a rigid-body velocity; see the book of Pozrikidis [111] for details but also [28] for a simple and relevant derivation using a regularized (nonsingular) Green's function. If there is slip, then $\boldsymbol{\lambda}$ does not have a direct physical interpretation as a surface traction; rather, it is the jump in the stress when going across the body surface from the "interior" flow to the "exterior" flow. If one wants to determine the actual traction in the presence of nontrivial slip, a second-kind integral formulation ought to be used, which includes an additional term on the right-hand side of (A-5) involving $\breve{\boldsymbol{u}}$ [111; 74; 119]. The fact that the same force and torque balance condition (A-3) applies even though $\boldsymbol{\lambda}$ is not the physical traction follows from the fact that the fictitious fluid inside the body is not accelerating; equivalently, one observes that a double-layer density does not contribute to the total force and torque on the body since it is a dipole rather than a monopole density. As discussed at length by Cortez et al. [28], both the method of regularized Stokeslets and the rigid multiblob method presented here can be seen as a particularly straightforward technique for solving a suitably regularized version of (A-5) [101; 120].

Appendix B: Computing flow fields

Observe that, unlike the immersed boundary method, the Green's-function-based rigid multiblob method described in Section 2 does not compute the actual flow (velocity and pressure) around the bodies. Rendering flow fields is useful in a number of applications for visualizing the flow around passive and active rigid bodies. There are a number of different ways to define a flow field around a multiblob; here we follow the following procedure that reuses existing code and produces smooth nonsingular flow fields everywhere, including inside the blobs. The input to the calculation is the constraint forces $\boldsymbol{\lambda}$, and the output is a fluid velocity $\boldsymbol{v}(\boldsymbol{r})$ evaluated at an arbitrary position in the domain.

The basic idea is to estimate the velocity that a freely moving tracer blob of a given size $a' \ll a$ would have, where a' is a desired resolution scale for the

flow that could be chosen to match the size of actual tracer particles used in an experiment. We replace each of the N_b blobs with N_b' smaller blobs of radius a'; i.e., we treat each blob i as a sphere of radius a and discretize it using smaller blobs. We divide the constraint force on blob i *uniformly* (this is consistent with the approximation used to construct the RPY tensor [130]) among the small blobs: $\boldsymbol{\lambda}_j' = \boldsymbol{\lambda}_i / N_b'$, where $j = 1, \ldots, N_b'$. The velocity field is then *defined* at an arbitrary point in space via $\boldsymbol{v}(\boldsymbol{r}) = \sum_j \boldsymbol{\mathcal{M}}'(\boldsymbol{r} - \boldsymbol{r}_j)\boldsymbol{\lambda}_j'$, where $\boldsymbol{\mathcal{M}}'$ is the blob-blob mobility for blobs of radius a'. Observe that the above sum can be evaluated using the existing matrix-vector product, but now applied to the collection of $N_b N_b'$ small blobs.

Appendix C: Permeable bodies: Brinkman equations

When the suspended rigid bodies are made of a porous material and thus partially permeable to the fluid, one can model the flow inside the particles using the (Debye–Bueche–)Brinkman [19; 40] equation, as done for suspensions of permeable spheres using multipole expansion methods in [1]. In this appendix we demonstrate analytically and numerically how a small change in the formulation can be used to allow for a finite permeability of the particles with minimal changes to the algorithm and implementation.

For particles with permeability (porosity) κ (possibly different for different bodies), the velocity equation extends to the whole domain including the interior of the bodies, and takes the form of Brinkman's equation

$$\nabla \pi = \eta \nabla^2 \boldsymbol{v} - \sum_p \frac{\eta}{\kappa_p} [\boldsymbol{v} - (\boldsymbol{u}_p + \boldsymbol{\omega}_p \times (\boldsymbol{r} - \boldsymbol{q}_p))] \mathbf{1}_p, \tag{C-1}$$

where $\mathbf{1}_p(\boldsymbol{r})$ is the characteristic function of body p, with the condition that both the velocity and the stress are continuous across the particle-fluid interface. Note that the rigid-body case corresponds to the limit $\kappa \to 0$ and is a singular limit in which the stress becomes discontinuous.

For permeable bodies, we fill the interior of the bodies with blobs as well, rather than just covering the surface with blobs as we did for impermeable bodies. Such filled rigid multiblob models can be constructed, for example, by covering the body with an unstructured tetrahedral grid with good uniformity properties and placing blobs at the nodes (vertices) of the grid. One also needs to assign a volume ΔV_i to each blob; this can be done by assigning $\frac{1}{4}$ of the volume of each tetrahedron to each of its 4 vertices. Once a *filled* rigid multiblob model of the body is constructed, the only change to the formulation (20) is to make the effective slip on blob i on body p proportional to the fluid-blob force:

$$\breve{\boldsymbol{u}}_i = -\frac{\kappa_p}{\eta \, \Delta V_i} \boldsymbol{\lambda}_i. \tag{C-2}$$

Note that this makes the system (20) strictly easier to solve than the case of impermeable bodies; the system is no longer a saddle-point problem for $\kappa > 0$. For the Green's-function-based methods described in Section 3, accounting for (C-2) simply adds $\kappa_p/(\eta\,\Delta V_i)$ to the diagonal elements $\mathcal{M}_{ii}$. For the Stokes-solver-based methods described in Section 3, all that is required is to set $\boldsymbol{\Omega}$ to be a diagonal matrix with $\Omega_{ii} = \kappa_p/(\eta\,\Delta V_i)$ for blob $i \in \mathcal{B}_p$ in (24).

To demonstrate that (C-2) is consistent with the Brinkman equations (C-1), we focus on the semicontinuum formulation (20). Solve the third equation in (20) for $\boldsymbol{\lambda}_i$ (note that this is only possible for nonzero permeability) and substitute the result into the first equation in (20) to obtain

$$\nabla\pi = \eta\nabla^2\boldsymbol{v} - \sum_p \frac{\eta}{\kappa_p} \sum_{i\in\mathcal{B}_p} \Delta V_i \times \left[\int \delta_a(\boldsymbol{r}_i - \boldsymbol{r}')\boldsymbol{v}(\boldsymbol{r}', t)\,d\boldsymbol{r}' - (\boldsymbol{u}_p + \boldsymbol{\omega}_p \times (\boldsymbol{r}_i - \boldsymbol{q}_p))\right]\delta_a(\boldsymbol{r}_i - \boldsymbol{r}). \quad \text{(C-3)}$$

In the limit in which the number of blobs goes to infinity and the regularized delta function δ_a becomes a true delta function, the sum over $i \in \mathcal{B}_p$ converges to

$$\int_{\Omega_p}\left[\int \delta(\boldsymbol{r}'' - \boldsymbol{r}')\boldsymbol{v}(\boldsymbol{r}', t)\,d\boldsymbol{r}' - (\boldsymbol{u}_p + \boldsymbol{\omega}_p \times (\boldsymbol{r}'' - \boldsymbol{q}_p))\right]\delta(\boldsymbol{r}'' - \boldsymbol{r})\,d\boldsymbol{r}'' \to [\boldsymbol{v}(\boldsymbol{r}) - (\boldsymbol{u}_p + \boldsymbol{\omega}_p \times (\boldsymbol{r} - \boldsymbol{q}_p))]\mathbf{1}_p(\boldsymbol{r})$$

and therefore the fluid equation (C-3) is a regularized semidiscretization of the Brinkman equation (C-1).

C.1. ***Numerical results.*** In this section we confirm the consistency of (C-2) with the Brinkman equations by comparing to analytical results. We also assess the accuracy of the method for different resolutions. Here we use the immersed boundary formulation, but we expect similar results to apply to methods based on analytical Green's functions.

Permeable slab. First, we compute the flow through a permeable slab to numerically estimate the effective permeability of a rigid multiblob for several spacings between the blobs. We compose a slab of thickness $5s$ from blobs placed on a cubic lattice with spacing s (i.e., the slab has 5 layers of blobs), and place the slab in the middle of a cubic domain with side L. We impose no slip for the tangential stress (traction) on all boundaries of the domain using the technique developed by Griffith [53]. For the normal component, on the sides of the domain perpendicular to the slab we impose no slip, and we impose a pressure jump of magnitude $\Delta\pi$ across the boundaries parallel to the slab. We measure the velocity U of the resulting nearly uniform flow (leak) through the slab as the velocity at the centerline of the slab a quarter of the domain from the left boundary.

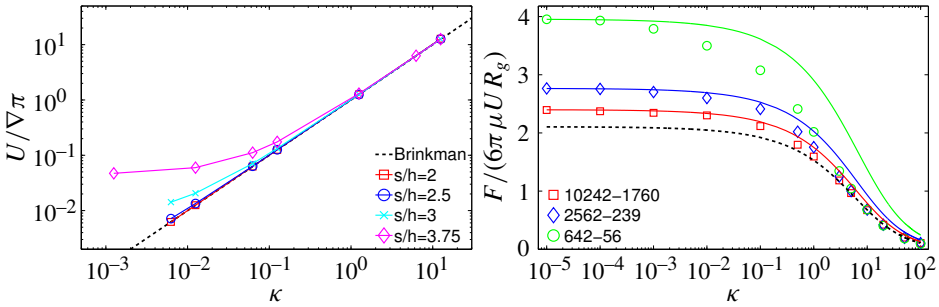


Figure 13. Left: numerically measured permeability of a slab as a function of the input permeability for different blob-blob spacings s. Right: measured drag (symbols) on a permeable sphere moving inside a fixed impermeable sphere, as a function of the input permeability, for 3 different resolutions, indicated as the number of blobs on the outer shell (642, 2562, or 10242 blobs) and inner sphere (56, 239, or 1760 blobs, respectively); see the legend. The theoretical result based on the geometric radii of the spheres is shown with a dashed black line, while the theoretical result based on the effective hydrodynamic radii in the impermeable limit (which vary with resolution) is shown with a solid line of the same color as the corresponding symbols.

At steady state we expect a uniform flow inside the slab with magnitude determined from the Brinkmann equation

$$\nabla\pi = \frac{\Delta\pi}{L} = -\frac{\eta}{\kappa}U, \tag{C-4}$$

where κ is the permeability (porosity). In the left panel of Figure 13 we compare this to the numerical observations. We see that for a variety of spacings between the blobs we get the correct permeability for large target values of κ. However, as we make the slab less and less permeable and approach the (singular) impermeable limit, we start to see a small but measurable leak in the rigid multiblob results. This leak is larger the larger the spacing between the blobs is, consistent with the intuition that leaking occurs between the blobs. This suggests that for permeable bodies it is better to reduce the spacing between the markers to $s \approx 2h$ as suggested in [71]. Note that the conditioning of the blob-blob mobility matrix is significantly improved for permeable bodies compared to impermeable bodies, so that this reduction in the spacing does not lead to conditioning problems except for very small values of κ.

Permeable sphere. Next we examine the translational mobility of a permeable sphere of radius a. The drag force on a permeable sphere of radius R moving through an unbounded domain with velocity U is given in [59; 45]:

$$(6\pi\eta U R)^{-1} F = \frac{G}{1+3G/(2\sigma^2)} = 1 - \frac{1}{\sigma} + O\Big(\frac{1}{\sigma^2}\Big),$$

where $G = 1 - \sigma^{-1}\tanh\sigma$ and $\sigma = \sqrt{a^2/\kappa}$. To eliminate finite-size effects, and following our prior work [71] for impermeable spheres, we consider here a permeable sphere inside an impermeable spherical shell, that is, we impose a no-slip boundary condition on a spherical shell of radius $b = a/\lambda$ that is concentric to the permeable sphere.

The equations we need to solve are the Stokes equations in the region between the spherical shells and the Brinkman equation (C-1) inside the permeable sphere, with no-slip boundary conditions on the outer shell and continuity of both velocity *and* stress on the boundary of the permeable sphere. The drag force can be shown to be

$$\alpha(6\pi\eta U R)^{-1}F = 5\lambda^5 + G(\sigma)\Big[1 - \lambda^5\Big(6 + \frac{15}{\sigma^2}\Big)\Big], \tag{C-5}$$

where

$$\alpha = 1 - 5\lambda^3 + \lambda^5\Big(9 + \frac{15}{2\sigma^2}\Big) - 5\lambda^6$$
$$+G(\sigma)\Big[\frac{3}{2\sigma^2} - \frac{9}{4}\lambda + \frac{15}{2}\lambda^3\Big(1 + \frac{1}{\sigma^2}\Big) - \frac{9}{2}\lambda^5\Big(\frac{5}{2} + \frac{7}{\sigma^2} + \frac{5}{\sigma^4}\Big) + \lambda^6\Big(6 + \frac{15}{\sigma^2}\Big)\Big]. \tag{C-6}$$

The solution to this problem in the limit of an impermeable sphere has been computed by Brenner [56] (see Appendix C in [71] for the full solution):

$$(6\pi\eta U R)^{-1}F = \frac{1 - \lambda^5}{1 - \frac{9}{4}\lambda + \frac{5}{2}\lambda^3 - \frac{9}{4}\lambda^5 + \lambda^6}. \tag{C-7}$$

The outside impermeable fixed shell is constructed as a rigid multiblob using the same recursive triangulation as before. Recall that the inner sphere has to be uniformly filled with blobs for $\kappa > 0$; we construct a filled-sphere model with typical spacing between nearest-neighbor blobs of $s \approx 2h$ using a tetrahedral mesh generator, starting from a uniform surface triangulation. In the right panel of Figure 13 we show the drag on the inner sphere compared to the theory (C-5), for several different resolutions. We observe that for large permeabilities there is an excellent agreement with the theory based on the *geometric* radii of the inner and outer spheres, even for rather modest resolutions. But for small permeabilities, we see deviations from the theory. This is not unexpected, since in the limit of zero permeability we must converge back to the rigid-sphere case, and we know that in this case the drag is determined by the larger effective *hydrodynamic* and not the geometric radius. Of course, as the resolution is refined we get convergence of the geometric and hydrodynamic radii, but convergence is very slow.

Our numerical observations are consistent with physical intuition. For large permeability, the flow is smooth and there is no jump in the pressure (and velocity derivatives) across the surface of the body, making the rigid multiblob models relatively accurate even for modest resolutions. However, for impermeable bodies

	μ^0	δ	α	$f_{n,2}$	$f_{n,1}$	$f_{n,0}$	$f_{d,1}$	$f_{d,0}$	error
$\mu^{\parallel}_{tt}$	$6\pi\eta R_h$	1	1	$-\frac{9}{16}$	0.826024	−0.311607	−1.4297	0.498974	$5.6\cdot10^{-3}$
$\mu^{\parallel}_{rr}$	$8\pi\eta R_\tau^3$	1	3	$-\frac{5}{16}$	0.15118	0.0830598	−0.443529	−0.406958	$4.9\cdot10^{-4}$
$\mu^{\perp}_{rr}$	$8\pi\eta R_\tau^3$	1	3	$-\frac{1}{8}$	0.122506	−0.0105777	−0.953632	0.0339739	$7.2\cdot10^{-5}$
$\mu^{\parallel}_{rt}$	$6\pi\eta R_h^2$	0	4	$\frac{3}{32}$	−0.142813	0.0508471	−0.528495	−0.454638	$6.6\cdot10^{-3}$

Table 8. Fitting parameters for the mobility of a sphere close to a wall obtained using the numerical mobility of the 642-blob model. In the last column we report the maximum relative error between the numerical mobility and the fit in the interval $(1.03, 10)R_h$.

the flow develops a boundary layer near the surface of the inner sphere and the pressure and velocity are no longer sufficiently smooth and the accuracy is lowered. We were able to account for the smearing of the no-slip condition for an impermeable (passive) sphere by adjusting the hydrodynamic radius $R_h > R_g$, but this adjustment cannot be made uniformly for all permeabilities. This is similar to the situation for active spheres discussed in Section 4.3, and highlights the inherent accuracy limitations of regularized methods, including both the rigid multiblob method and the method of regularized Stokeslets.

Appendix D: Empirical mobility of a sphere near a wall

A number of theoretical predictions are available for the mobility of a sphere close to a wall [56] (see Appendix D in [34] for a summary). However, except for the translational mobility perpendicular to the wall, for which Brenner computed an exact infinite sum [17] (see [63] for an approximate rational fit), the theoretical results are based on asymptotic expansions and have a limited range of validity. Since the dynamics of spherical colloids near a no-slip boundary is relevant to a number of experimental studies, we give here empirical fits to the mobility computed in Section 5.1 using a rigid multiblob with 642 blobs (our highest resolution).

We have fitted the mobilities shown in the panels of Figure 5 with a rational function of the form

$$\frac{\mu(x = H/R_h)}{\mu^0} = \delta + \left(\frac{1}{x}\right)^{\alpha} \frac{f_{n,2}x^2 + f_{n,1}x + f_{n,0}}{x^2 + f_{d,1}x + f_{d,0}},$$

where μ^0 is the bulk mobility and δ, α, and $f_{n,2}$ have been chosen to ensure the correct leading-order asymptotic scaling for large distances to the wall H and the rest of the constants are fitting parameters. The values of all the coefficients are given in Table 8.

References

[1] G. C. Abade, B. Cichocki, M. L. Ekiel-Jeżewska, G. Nägele, and E. Wajnryb, *Short-time dynamics of permeable particles in concentrated suspensions*, J. Chem. Phys. **132** (2010), no. 1, 014503.

[2] J. Ainley, S. Durkin, R. Embid, P. Boindala, and R. Cortez, *The method of images for regularized Stokeslets*, J. Comput. Phys. **227** (2008), no. 9, 4600–4616.

[3] P. J. Atzberger, *A note on the correspondence of an immersed boundary method incorporating thermal fluctuations with Stokesian–Brownian dynamics*, Phys. D **226** (2007), no. 2, 144–150.

[4] P. J. Atzberger, *Stochastic Eulerian Lagrangian methods for fluid-structure interactions with thermal fluctuations*, J. Comput. Phys. **230** (2011), no. 8, 2821–2837.

[5] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, *Efficient management of parallelism in object-oriented numerical software libraries*, Modern software tools for scientific computing (E. Arge, A. M. Bruaset, and H. P. Langtangen, eds.), Birkhäuser, Boston, 1997, Software available at http://www.mcs.anl.gov/petsc, pp. 163–202.

[6] F. Balboa Usabiaga, J. B. Bell, R. Delgado-Buscalioni, A. Donev, T. G. Fai, B. E. Griffith, and C. S. Peskin, *Staggered schemes for fluctuating hydrodynamics*, Multiscale Model. Simul. **10** (2012), no. 4, 1369–1408.

[7] F. Balboa Usabiaga, R. Delgado-Buscalioni, B. E. Griffith, and A. Donev, *Inertial coupling method for particles in an incompressible fluctuating fluid*, Comput. Methods Appl. Mech. Engrg. **269** (2014), 139–172, Software available at https://github.com/fbusabiaga/fluam.

[8] A. J. Banchio and J. F. Brady, *Accelerated Stokesian dynamics: Brownian motion*, J. Chem. Phys. **118** (2003), no. 22, 10323.

[9] Y. Bao, J. Kaye, and C. S. Peskin, *A Gaussian-like immersed-boundary kernel with three continuous derivatives and improved translational invariance*, J. Comput. Phys. **316** (2016), 139–144, Software available at https://github.com/stochasticHydroTools/IBMethod.

[10] C. W. J. Beenakker, *Ewald sum of the Rotne–Prager tensor*, J. Chem. Phys. **85** (1986), no. 3, 1581–1582.

[11] T. Benesch, S. Yiacoumi, and C. Tsouris, *Brownian motion in confinement*, Phys. Rev. E **68** (2003), no. 2, 021401.

[12] S. Bhattacharya, J. Bławzdziewicz, and E. Wajnryb, *Hydrodynamic interactions of spherical particles in suspensions confined between two planar walls*, J. Fluid Mech. **541** (2005), 263–292.

[13] J. R. Blake, *A note on the image system for a stokeslet in a no-slip boundary*, Math. Proc. Cambridge **70** (1971), no. 2, 303–310.

[14] A. Bouras and V. Frayssé, *Inexact matrix-vector products in Krylov methods for solving linear systems: a relaxation strategy*, SIAM J. Matrix Anal. Appl. **26** (2005), no. 3, 660–678.

[15] J. F. Brady and L. J. Durlofsky, *The sedimentation rate of disordered suspensions*, Phys. Fluids **31** (1988), no. 4, 717–727.

[16] J. F. Brady, R. J. Phillips, J. C. Lester, and G. Bossis, *Dynamic simulation of hydrodynamically interacting suspensions*, J. Fluid Mech. **195** (1988), 257–280.

[17] H. Brenner, *The slow motion of a sphere through a viscous fluid towards a plane surface*, Chem. Eng. Sci. **16** (1961), no. 3–4, 242–251.

[18] T. T. Bringley and C. S. Peskin, *Validation of a simple method for representing spheres and slender bodies in an immersed boundary method for Stokes flow on an unbounded domain*, J. Comput. Phys. **227** (2008), no. 11, 5397–5425.

[19] H. C. Brinkman, *A calculation of the viscous force exerted by a flowing fluid on a dense swarm of particles*, Appl. Sci. Res. **1** (1949), 27–34.

[20] M. Cai, A. Nonaka, J. B. Bell, B. E. Griffith, and A. Donev, *Efficient variable-coefficient finite-volume Stokes solvers*, Commun. Comput. Phys. **16** (2014), no. 5, 1263–1297.

[21] A. Chakrabarty, A. Konya, F. Wang, J. V. Selinger, K. Sun, and Q.-H. Wei, *Brownian motion of boomerang colloidal particles*, Phys. Rev. Lett. **111** (2013), no. 16, 160603.

[22] A. Chakrabarty, A. Konya, F. Wang, J. V. Selinger, K. Sun, and Q.-H. Wei, *Brownian motion of arbitrarily shaped particles in two dimensions*, Langmuir **30** (2014), no. 46, 13844–13853.

[23] A. Chakrabarty, F. Wang, C.-Z. Fan, K. Sun, and Q.-H. Wei, *High-precision tracking of Brownian boomerang colloidal particles confined in quasi two dimensions*, Langmuir **29** (2013), no. 47, 14396–14402.

[24] B. Cichocki, B. U. Felderhof, K. Hinsen, E. Wajnryb, and J. Bławzdziewicz, *Friction and mobility of many spheres in Stokes flow*, J. Chem. Phys. **100** (1994), no. 5, 3780–3790.

[25] B. Cichocki and K. Hinsen, *Stokes drag on conglomerates of spheres*, Phys. Fluids **7** (1995), no. 2, 285–291.

[26] B. Cichocki, R. B. Jones, R. Kutteh, and E. Wajnryb, *Friction and mobility for colloidal spheres in Stokes flow near a boundary: the multipole method and applications*, J. Chem. Phys. **112** (2000), no. 5, 2548–2560.

[27] R. Cortez, *The method of regularized Stokeslets*, SIAM J. Sci. Comput. **23** (2001), no. 4, 1204–1225.

[28] R. Cortez, L. Fauci, and A. Medovikov, *The method of regularized Stokeslets in three dimensions: analysis, validation, and application to helical swimming*, Phys. Fluids **17** (2005), no. 3, 031504.

[29] M. S. Davies Wykes, J. Palacci, T. Adachi, L. Ristroph, X. Zhong, M. D. Ward, J. Zhang, and M. J. Shelley, *Dynamic self-assembly of microscale rotors and swimmers*, Soft Matter **12** (2016), no. 20, 4584–4589.

[30] B. Delmotte, E. Climent, and F. Plouraboué, *A general formulation of bead models applied to flexible fibers and active filaments at low Reynolds number*, J. Comput. Phys. **286** (2015), 14–37.

[31] B. Delmotte and E. E. Keaveny, *Simulating Brownian suspensions with fluctuating hydrodynamics*, J. Chem. Phys. **143** (2015), no. 24, 244109.

[32] B. Delmotte, E. E. Keaveny, F. Plouraboué, and E. Climent, *Large-scale simulation of steady and time-dependent active suspensions with the force-coupling method*, J. Comput. Phys. **302** (2015), 524–547.

[33] S. Delong, F. Balboa Usabiaga, R. Delgado-Buscalioni, B. E. Griffith, and A. Donev, *Brownian dynamics without Green's functions*, J. Chem. Phys. **140** (2014), no. 13, 134110, Software available at https://github.com/stochasticHydroTools/FIB.

[34] S. Delong, F. Balboa Usabiaga, and A. Donev, *Brownian dynamics of confined rigid bodies*, J. Chem. Phys. **143** (2015), no. 14, 144107, Software available at https://github.com/stochasticHydroTools/RotationalDiffusion.

[35] M. Długosz and J. M. Antosiewicz, *Toward an accurate modeling of hydrodynamic effects on the translational and rotational dynamics of biomolecules in many-body systems*, J. Phys. Chem. B **119** (2015), no. 26, 8425–8439.

[36] A. Donev, J. Burton, F. H. Stillinger, and S. Torquato, *Tetratic order in the phase behavior of a hard-rectangle system*, Phys. Rev. B **73** (2006), no. 5, 054109.

[37] A. Donev, S. Torquato, and F. H. Stillinger, *Neighbor list collision-driven molecular dynamics simulation for nonspherical hard particles, I: Algorithmic details*, J. Comput. Phys. **202** (2005), no. 2, 737–764, Software available at http://cims.nyu.edu/~donev/Packing/PackLSD/Instructions.html.

[38] A. Donev, S. Torquato, and F. H. Stillinger, *Neighbor list collision-driven molecular dynamics simulation for nonspherical hard particles, II: Applications to ellipses and ellipsoids*, J. Comput. Phys. **202** (2005), no. 2, 765–793, Software available at http://cims.nyu.edu/~donev/Packing/PackLSD/Instructions.html.

[39] A. Donev and E. Vanden-Eijnden, *Dynamic density functional theory with hydrodynamic interactions and fluctuations*, J. Chem. Phys. **140** (2014), no. 23, 234115.

[40] L. Durlofsky and J. F. Brady, *Analysis of the Brinkman equation as a model for flow in porous media*, Phys. Fluids **30** (1987), no. 11, 3329–3341.

[41] H. Elman, V. E. Howle, J. Shadid, R. Shuttleworth, and R. Tuminaro, *Block preconditioners based on approximate commutators*, SIAM J. Sci. Comput. **27** (2006), no. 5, 1651–1668.

[42] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*, 2nd ed., Oxford University, 2014.

[43] R. D. Falgout, J. E. Jones, and U. M. Yang, *The design and implementation of hypre, a library of parallel high performance preconditioners*, Numerical solution of partial differential equations on parallel computers (A. M. Bruaset and A. Tveito, eds.), Lect. Notes Comput. Sci. Eng., no. 51, Springer, Berlin, 2006, Software available at http://www.llnl.gov/CASC/hypre, pp. 267–294.

[44] L. P. Faucheux and A. J. Libchaber, *Confined Brownian motion*, Phys. Rev. E **49** (1994), no. 6, 5158–5163.

[45] B. U. Felderhof, *Frictional properties of dilute polymer solutions, III: Translational-friction coefficient*, Physica A **80** (1975), no. 1, 63–75.

[46] M. X. Fernandes and J. García de la Torre, *Brownian dynamics simulation of rigid particles of arbitrary shape in external fields*, Biophys. J. **83** (2002), no. 6, 3039–3048.

[47] J. M. García Bernal and J. García de la Torre, *Transport properties and hydrodynamic centers of rigid macromolecules with arbitrary shapes*, Biopolymers **19** (1980), no. 4, 751–766.

[48] J. García de la Torre, M. L. Huertas, and B. Carrasco, *Calculation of hydrodynamic properties of globular proteins from their atomic-level structure*, Biophys. J. **78** (2000), no. 2, 719–730.

[49] E. M. Gauger, M. T. Downton, and H. Stark, *Fluid transport at low Reynolds number with magnetically actuated artificial cilia*, Eur. Phys. J. E **28** (2009), no. 2, 231–242.

[50] S. Ghose and R. Adhikari, *Irreducible representations of oscillatory and swirling flows in active soft matter*, Phys. Rev. Lett. **112** (2014), no. 11, 118102.

[51] Z. Gimbutas, L. Greengard, and S. Veerapaneni, *Simple and efficient representations for the fundamental solutions of Stokes flow in a half-space*, J. Fluid Mech. **776** (2015), R1, Software available at http://www.cims.nyu.edu/cmcl/fmm3dlib/fmm3dlib.html.

[52] A. J. Goldman, R. G. Cox, and H. Brenner, *Slow viscous motion of a sphere parallel to a plane wall, I: Motion through a quiescent fluid*, Chem. Eng. Sci. **22** (1967), no. 4, 637–651.

[53] B. E. Griffith, *An accurate and efficient method for the incompressible Navier–Stokes equations using the projection method as a preconditioner*, J. Comput. Phys. **228** (2009), no. 20, 7565–7595.

[54] B. E. Griffith, R. D. Hornung, D. M. McQueen, and C. S. Peskin, *An adaptive, formally second order accurate version of the immersed boundary method*, J. Comput. Phys. **223** (2007), no. 1, 10–49, Software available at https://github.com/ibamr/ibamr.

[55] B. E. Griffith and X. Luo, *Hybrid finite difference/finite element version of the immersed boundary method*, preprint, 2016.

[56] J. Happel and H. Brenner, *Low reynolds number hydrodynamics: with special applications to particulate media*, Mech. Fluids Transport Processes, no. 1, Martinus Nijhoff, The Hague, 1983.

[57] J. P. Hernández-Ortiz, J. J. de Pablo, and M. D. Graham, *N log N method for hydrodynamic interactions of confined polymer systems: Brownian dynamics*, J. Chem. Phys. **125** (2006), no. 16, 164906.

[58] J. P. Hernández-Ortiz, J. J. de Pablo, and M. D. Graham, *Fast computation of many-particle hydrodynamic and electrostatic interactions in a confined geometry*, Phys. Rev. Lett. **98** (2007), no. 14, 140602.

[59] J. J. L. Higdon and M. Kojima, *On the calculation of Stokes' flow past porous particles*, Int. J. Multiphase Flow **7** (1981), no. 6, 719–727.

[60] R. J. Hill, D. L. Koch, and A. J. C. Ladd, *Moderate-Reynolds-number flows in ordered and random arrays of spheres*, J. Fluid Mech. **448** (2001), 243–278.

[61] E. J. Hinch, *Application of the Langevin equation to fluid suspensions*, J. Fluid Mech. **72** (1975), no. 3, 499–511.

[62] K. Hinsen, *HYDROLIB: a library for the evaluation of hydrodynamic interactions in colloidal suspensions*, Comput. Phys. Commun. **88** (1995), no. 2–3, 327–340.

[63] P. Huang and K. S. Breuer, *Direct measurement of anisotropic near-wall hindered diffusion using total internal reflection velocimetry*, Phys. Rev. E **76** (2007), no. 4, 046307.

[64] M. Hütter and H. C. Öttinger, *Fluctuation-dissipation theorem, kinetic stochastic integral and efficient simulations*, J. Chem. Soc. Farad. T. **94** (1998), no. 10, 1403–1405.

[65] K. Ichiki, *Improvement of the Stokesian dynamics method for systems with a finite number of particles*, J. Fluid Mech. **452** (2002), 231–262, Software available at http://kichiki.github.com/libstokes/.

[66] D. J. Jeffrey and Y. Onishi, *The slow motion of a cylinder next to a plane wall*, Quart. J. Mech. Appl. Math. **34** (1981), no. 2, 129–137.

[67] D. J. Jeffrey and Y. Onishi, *Calculation of the resistance and mobility functions for two unequal rigid spheres in low-Reynolds-number flow*, J. Fluid Mech. **139** (1984), 261–290.

[68] R. M. Jendrejack, J. J. de Pablo, and M. D. Graham, *Stochastic simulations of DNA in flow: dynamics and the effects of hydrodynamic interactions*, J. Chem. Phys. **116** (2002), no. 117, 7752–7759.

[69] R. M. Jendrejack, D. C. Schwartz, M. D. Graham, and J. J. de Pablo, *Effect of confinement on DNA dynamics in microfluidic devices*, J. Chem. Phys. **119** (2003), no. 2, 1165–1173.

[70] H. Jiang and Z. Hou, *Hydrodynamic interaction induced spontaneous rotation of coupled active filaments*, Soft Matter **10** (2014), no. 46, 9248–9253.

[71] B. Kallemov, A. P. S. Bhalla, B. E. Griffith, and A. Donev, *An immersed boundary method for rigid bodies*, Commun. Appl. Math. Comput. Sci. **11** (2016), no. 1, 79–141, Software available at https://github.com/stochasticHydroTools/RigidBodyIB.

[72] D. F. Katz, J. R. Blake, and S. L. Paveri-Fontana, *On the movement of slender bodies near plane boundaries at low Reynolds number*, J. Fluid Mech. **72** (1975), no. 3, 529–540.

[73] E. E. Keaveny, *Fluctuating force-coupling method for simulations of colloidal suspensions*, J. Comput. Phys. **269** (2014), 61–79.

[74] E. E. Keaveny and M. J. Shelley, *Applying a second-kind boundary integral equation for surface tractions in Stokes flow*, J. Comput. Phys. **230** (2011), no. 5, 2141–2159.

[75] R. Kekre, J. E. Butler, and A. J. C. Ladd, *Comparison of lattice-Boltzmann and Brownian-dynamics simulations of polymer migration in confined flows*, Phys. Rev. E **82** (2010), no. 1, 011802.

[76] J. B. Keller and S. I. Rubinow, *Slender-body theory for slow viscous flow*, J. Fluid Mech. **75** (1976), no. 4, 705–714.

[77] L. af Klinteberg and A.-K. Tornberg, *Fast Ewald summation for Stokesian particle suspensions*, Internat. J. Numer. Methods Fluids **76** (2014), no. 10, 669–698.

[78] A. Klöckner, N. Pinto, Y. Lee, B. Catanzaro, P. Ivanov, and A. Fasih, *PyCUDA and PyOpenCL: a scripting-based approach to GPU run-time code generation*, Parallel Comput. **38** (2012), no. 3, 157–174.

[79] D. L. Koch and G. Subramanian, *Collective hydrodynamics of swimming microorganisms: living fluids*, Annu. Rev. Fluid Mech. **43** (2011), 637–659.

[80] R. Kutteh, *Rigid body dynamics approach to Stokesian dynamics simulations of nonspherical particles*, J. Chem. Phys. **132** (2010), no. 17, 174107.

[81] A. J. C. Ladd, *Hydrodynamic interactions in a suspension of spherical particles*, J. Chem. Phys. **88** (1988), no. 8, 5051–5063.

[82] A. J. C. Ladd, *Hydrodynamic transport coefficients of random dispersions of hard spheres*, J. Chem. Phys. **93** (1990), no. 5, 3484–3494.

[83] A. J. C. Ladd, *Dynamical simulations of sedimenting spheres*, Phys. Fluids A **5** (1993), no. 2, 299–310.

[84] A. J. C. Ladd, *Numerical simulations of particulate suspensions via a discretized Boltzmann equation, II: Numerical results*, J. Fluid Mech. **271** (1994), 311–339.

[85] A. J. C. Ladd, *Sedimentation of homogeneous suspensions of non-Brownian spheres*, Phys. Fluids **9** (1997), no. 3, 491–499.

[86] A. J. C. Ladd, R. Kekre, and J. E. Butler, *Comparison of the static and dynamic properties of a semiflexible polymer using lattice Boltzmann and Brownian-dynamics simulations*, Phys. Rev. E **80** (2009), no. 3, 036704.

[87] M. H. Langston, L. Greengard, and D. Zorin, *A free-space adaptive FMM-based PDE solver in three dimensions*, Commun. Appl. Math. Comput. Sci. **6** (2011), no. 1, 79–122, Software available at https://github.com/dmalhotra/pvfmm.

[88] K. Leiderman, E. L. Bouzarth, R. Cortez, and A. T. Layton, *A regularization method for the numerical solution of periodic Stokes flow*, J. Comput. Phys. **236** (2013), 187–202.

[89] Z. Liang, Z. Gimbutas, L. Greengard, J. Huang, and S. Jiang, *A fast multipole method for the Rotne–Prager–Yamakawa tensor and its applications*, J. Comput. Phys. **234** (2013), 133–139.

[90] D. Lindbo and A.-K. Tornberg, *Spectrally accurate fast summation for periodic Stokes potentials*, J. Comput. Phys. **229** (2010), no. 23, 8994–9010.

[91] V. Lobaskin and B. Dünweg, *A new model for simulating colloidal dynamics*, New J. Phys. **6** (2004), 54.

[92] S. Lomholt and M. R. Maxey, *Force-coupling method for particulate two-phase flow: Stokes flow*, J. Comput. Phys. **184** (2003), no. 2, 381–405.

[93] E. Lushi and C. S. Peskin, *Modeling and simulation of active suspensions containing large numbers of interacting micro-swimmers*, Comput. Struct. **122** (2013), 239–248.

[94] F. Ma, X. Yang, H. Zhao, and N. Wu, *Inducing propulsion of colloidal dimers by breaking the symmetry in electrohydrodynamic flow*, Phys. Rev. Lett. **115** (2015), no. 20, 208302.

[95] T. Majmudar, E. E. Keaveny, J. Zhang, and M. J. Shelley, *Experiments and theory of undulatory locomotion in a simple structured medium*, J. R. Soc. Interface **9** (2012), no. 73, 1809–1823.

[96] O. Marin, K. Gustavsson, and A.-K. Tornberg, *A highly accurate boundary treatment for confined Stokes flow*, Comput. Fluids **66** (2012), 215–230.

[97] M. R. Maxey and B. K. Patel, *Localized force representations for particles sedimenting in Stokes flow*, Int. J. Multiphase Flow **27** (2001), no. 9, 1603–1626.

[98] N. J. de Mestre and W. B. Russel, *Low-Reynolds-number translation of a slender cylinder near a plane wall*, J. Eng. Math. **9** (1975), no. 2, 81–91.

[99] K. A. Mizerski, E. Wajnryb, P. J. Zuk, and P. Szymczak, *The Rotne–Prager–Yamakawa approximation for periodic systems in a shear flow*, J. Chem. Phys. **140** (2014), no. 18, 184103.

[100] J. J. Molina and R. Yamamoto, *Direct numerical simulations of rigid body dispersions, I: Mobility/friction tensors of assemblies of spheres*, J. Chem. Phys. **139** (2013), no. 23, 234105.

[101] T. D. Montenegro-Johnson, S. Michelin, and E. Lauga, *A regularised singularity approach to phoretic problems*, Eur. Phys. J. E **38** (2015), 139.

[102] E. Nazockdast, A. Rahimian, D. Zorin, and M. Shelley, *A fast platform for simulating semi-flexible fiber suspensions applied to cell mechanics*, J. Comput. Phys **329** (2017), 173–209.

[103] N.-Q. Nguyen and A. J. C. Ladd, *Sedimentation of hard-sphere suspensions at low Reynolds number*, J. Fluid Mech. **525** (2005), 72–104.

[104] A. Ortega, D. Amorós, and J. García de la Torre, *Prediction of hydrodynamic and other solution properties of rigid proteins from atomic- and residue-level models*, Biophys. J. **101** (2011), no. 4, 892–898, Software available at http://leonardo.inf.um.es/macromol/programs/hydropro/hydropro.htm.

[105] J. Palacci, S. Sacanna, A. P. Steinberg, D. J. Pine, and P. M. Chaikin, *Living crystals of light-activated colloidal surfers*, Science **339** (2013), no. 6122, 936–940.

[106] A. Pandey, P. B. S. Kumar, and R. Adhikari, *Flow-induced nonequilibrium self-assembly in suspensions of stiff, apolar, active filaments*, Soft Matter **12** (2016), no. 44, 9068–9076.

[107] C. S. Peskin, *The immersed boundary method*, Acta Numer. **11** (2002), 479–517.

[108] T. T. Pham, U. D. Schiller, J. R. Prakash, and B. Dünweg, *Implicit and explicit solvent models for the simulation of a single polymer chain in solution: lattice Boltzmann versus Brownian dynamics*, J. Chem. Phys. **131** (2009), no. 16, 164114.

[109] P. Plunkett, J. Hu, C. Siefert, and P. J. Atzberger, *Spatially adaptive stochastic methods for fluid-structure interactions subject to thermal fluctuations in domains with complex geometries*, J. Comput. Phys. **277** (2014), 121–137.

[110] S. Poblete, A. Wysocki, G. Gompper, and R. G. Winkler, *Hydrodynamics of discrete-particle models of spherical colloids: a multiparticle collision dynamics simulation study*, Phys. Rev. E **90** (2014), no. 3, 033314.

[111] C. Pozrikidis, *Boundary integral and singularity methods for linearized viscous flow*, Cambridge University, 1992.

[112] A. M. Roma, C. S. Peskin, and M. J. Berger, *An adaptive version of the immersed boundary method*, J. Comput. Phys. **153** (1999), no. 2, 509–534.

[113] J. Rotne and S. Prager, *Variational treatment of hydrodynamic interaction in polymers*, J. Chem. Phys. **50** (1969), no. 11, 4831–4837.

[114] J.-N. Roux, *Brownian particles at different times scales: a new derivation of the Smoluchowski equation*, Phys. A **188** (1992), no. 4, 526–552.

[115] M. Rozložník and V. Simoncini, *Krylov subspace methods for saddle point problems with indefinite preconditioning*, SIAM J. Matrix Anal. Appl. **24** (2002), no. 2, 368–391.

[116] Y. Saa, *Iterative methods for sparse linear systems*, 2nd ed., SIAM, Philadelphia, 2003.

[117] A. Sierou and J. F. Brady, *Accelerated Stokesian Dynamics simulations*, J. Fluid Mech. **448** (2001), 115–146.

[118] R. Singh and R. Adhikari, *Universal hydrodynamic mechanisms for crystallization in active colloidal suspensions*, Phys. Rev. Lett. **117** (2016), no. 22, 228002.

[119] R. Singh, S. Ghose, and R. Adhikari, *Many-body microhydrodynamics of colloidal particles with active boundary layers*, J. Stat. Mech. Theory Exp. (2015), no. 6, P06017.

[120] D. J. Smith, *A boundary element regularized Stokeslet method applied to cilia- and flagella-driven flow*, Proc. R. Soc. Lond. A **465** (2009), no. 2112, 3605–3626.

[121] H. A. Stone and A. D. T. Samuel, *Propulsion of microorganisms by surface distortions*, Phys. Rev. Lett. **77** (1996), no. 19, 4102–4014.

[122] J. W. Swan and J. F. Brady, *Simulation of hydrodynamically interacting particles near a no-slip boundary*, Phys. Fluids **19** (2007), no. 11, 113306.

[123] J. W. Swan and J. F. Brady, *Particle motion between parallel walls: hydrodynamics and simulation*, Phys. Fluids **22** (2010), no. 10, 103301.

[124] J. W. Swan and J. F. Brady, *The hydrodynamics of confined dispersions*, J. Fluid Mech. **687** (2011), 254–299.

[125] J. W. Swan, J. F. Brady, R. S. Moore, and ChE 174, *Modeling hydrodynamic self-propulsion with Stokesian Dynamics: or teaching Stokesian Dynamics to swim*, Phys. Fluids **23** (2011), no. 7, 071901.

[126] J. W. Swan and G. Wang, *Rapid calculation of hydrodynamic and transport properties in concentrated solutions of colloidal particles and macromolecules*, Phys. Fluids **28** (2016), no. 1, 011902.

[127] D. Takagi, A. B. Braunschweig, J. Zhang, and M. J. Shelley, *Dispersion of self-propelled rods undergoing fluctuation-driven flips*, Phys. Rev. Lett. **110** (2013), no. 3, 038301.

[128] J. F. Trahan and R. G. Hussey, *The Stokes drag on a horizontal cylinder falling toward a horizontal plane*, Phys. Fluids **28** (1985), no. 10, 2961–2967.

[129] A. Vázquez-Quesada, F. Balboa Usabiaga, and R. Delgado-Buscalioni, *A multiblob approach to colloidal hydrodynamics with inherent lubrication*, J. Chem. Phys. **141** (2014), no. 20, 204102.

[130] E. Wajnryb, K. A. Mizerski, P. J. Zuk, and P. Szymczak, *Generalization of the Rotne–Prager–Yamakawa mobility and shear disturbance tensors*, J. Fluid Mech. **731** (2013), R3.

[131] M. Wang and J. F. Brady, *Short-time transport properties of bidisperse suspensions and porous media: a Stokesian dynamics study*, J. Chem. Phys. **142** (2015), no. 9, 094901.

[132] M. Wang and J. F. Brady, *Spectral Ewald acceleration of Stokesian dynamics for polydisperse suspensions*, J. Comput. Phys. **306** (2016), 443–477.

[133] T. Wang, S. K. Layton, and L. A. Barba, *Inexact Krylov iterations and relaxation strategies with fast-multipole boundary element method*, preprint, 2016, Software available at https://github.com/barbagroup/fmm-bem-relaxed.

[134] K. Yeo and M. R. Maxey, *Dynamics of concentrated suspensions of non-colloidal particles in Couette flow*, J. Fluid Mech. **649** (2010), 205–231.

[135] Y. Zhang, J. J. de Pablo, and M. D. Graham, *An immersed boundary method for Brownian dynamics simulation of polymers in complex geometries: application to DNA flowing through a nanoslit with embedded nanopits*, J. Chem. Phys. **136** (2012), no. 1, 014901.

[136] M. Zurita-Gotor, J. Bławzdziewicz, and E. Wajnryb, *Motion of a rod-like particle between parallel walls with application to suspension rheology*, J. Rheol. **51** (2007), no. 1, 71–97.

Received June 7, 2016. Revised November 20, 2016.

FLORENCIO BALBOA USABIAGA: fbalboa@courant.nyu.edu
Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, United States

BAKYTZHAN KALLEMOV: bkallemov@lbl.gov
Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, United States

and

Energy Geosciences Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, United States

BLAISE DELMOTTE: delmotte@cims.nyu.edu
Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, United States

AMNEET PAL SINGH BHALLA: amneetb@live.unc.edu
Department of Mathematics, University of North Carolina, Chapel Hill, NC 27599, United States

BOYCE E. GRIFFITH: boyceg@gmail.com
Department of Mathematics, University of North Carolina, Chapel Hill, NC 27599, United States

and

Department of Biomedical Engineering, University of North Carolina, Chapel Hill, NC 27599, United States

ALEKSANDAR DONEV: donev@courant.nyu.edu
Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, United States

Guidelines for Authors

Authors may submit manuscripts in PDF format on-line at the Submission page at msp.org/camcos.

Originality. Submission of a manuscript acknowledges that the manuscript is original and and is not, in whole or in part, published or under consideration for publication elsewhere. It is understood also that the manuscript will not be submitted elsewhere while under consideration for publication in this journal.

Language. Articles in CAMCoS are usually in English, but articles written in other languages are welcome.

Required items. A brief abstract of about 150 words or less must be included. It should be self-contained and not make any reference to the bibliography. If the article is not in English, two versions of the abstract must be included, one in the language of the article and one in English. Also required are keywords and subject classifications for the article, and, for each author, postal address, affiliation (if appropriate), and email address.

Format. Authors are encouraged to use LaTeX but submissions in other varieties of TeX, and exceptionally in other formats, are acceptable. Initial uploads should be in PDF format; after the refereeing process we will ask you to submit all source material.

References. Bibliographical references should be complete, including article titles and page ranges. All references in the bibliography should be cited in the text. The use of BibTeX is preferred but not required. Tags will be converted to the house format, however, for submission you may use the format of your choice. Links will be provided to all literature with known web locations and authors are encouraged to provide their own links in addition to those supplied in the editorial process.

Figures. Figures must be of publication quality. After acceptance, you will need to submit the original source files in vector graphics format for all diagrams in your manuscript: vector EPS or vector PDF files are the most useful.

Most drawing and graphing packages (Mathematica, Adobe Illustrator, Corel Draw, MATLAB, etc.) allow the user to save files in one of these formats. Make sure that what you are saving is vector graphics and not a bitmap. If you need help, please write to graphics@msp.org with details about how your graphics were generated.

White space. Forced line breaks or page breaks should not be inserted in the document. There is no point in your trying to optimize line and page breaks in the original manuscript. The manuscript will be reformatted to use the journal's preferred fonts and layout.

Proofs. Page proofs will be made available to authors (or to the designated corresponding author) at a Web site in PDF format. Failure to acknowledge the receipt of proofs or to return corrections within the requested deadline may cause publication to be postponed.

Communications in Applied Mathematics and Computational Science

vol. 11 no. 2 2016