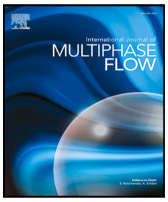
# A nonlocal multiscale model for Brownian particles: Application to hindered deposition in microfluidic systems

Citation for the original published paper (version of record):
Michael, A., Mark, A., Sasic, S. et al (2025). A nonlocal multiscale model for Brownian particles: Application to hindered deposition in
microfluidic systems. International Journal of Multiphase Flow, 193.
http://dx.doi.org/10.1016/j.ijmultiphaseflow.2025.105421

Contents lists available at ScienceDirect

International Journal of Multiphase Flow

journal homepage: www.elsevier.com/locate/ijmulflow



Research paper

# A nonlocal multiscale model for Brownian particles: Application to hindered deposition in microfluidic systems

Anand Joseph Michael [a,*], Andreas Mark [b], Srdjan Sasic [a], Henrik Ström [a]

[a] Division of Fluid Dynamics, Chalmers University of Technology, Gothenburg, SE-412 96, Sweden

[b] Department of Computational Engineering and Design, Fraunhofer-Chalmers Research Centre, Gothenburg, SE-412 88, Sweden

ARTICLE INFO

*Keywords:*
Colored Brownian motion
Memory kernel
Generalized Langevin equation
Lagrangian Particle Tracking
Multiphase Direct Numerical Simulation

ABSTRACT

Separation of neutrally buoyant sub-micron particles from a fluid flow in a microfluidic device under the action of an external field (e.g. electric, magnetic, gravitational, concentration) is an important challenge in the engineering sciences. In such processes, the hydrodynamic force resisting the motion of the particles towards a target wall becomes distinctively nonlocal, being influenced by particle velocities in past locations via the history force. The nonlocality of the hydrodynamic force implies that the Brownian force exhibits similar correlations in time, owing to the fluctuation–dissipation relation. The problem of identifying the optimal design parameters for the microfluidic system then traditionally necessitates the application of computationally expensive simulation models.

In this work, we develop a novel computationally efficient nonlocal multiscale model for Brownian particles in confined geometries. The model accounts for nonlocality and hindrance effects by employing multiphase Direct Numerical Simulation (DNS) data to determine the memory kernels associated with the particles in the system. The memory kernels are then used in a Lagrangian Particle Tracking (LPT) routine to evolve particle trajectories without the need for analytical models to describe the forces involved and their modulation due to hindrance.

We show that the focusing of particles on a target wall during hindered deposition in microfluidic systems can be controlled via two parameters (the magnitude of the attracting force and the size of the geometry) that interact in non-trivial ways. For the parameter space investigated in the current work, the focusing, as quantified via the normalized impact radius, can be reduced by up to 80% by optimally designing the microfluidic device. The implications of using the new model for further development of microfluidic particulate systems are discussed.

## 1. Introduction

Microfluidic systems have emerged over the past decades as a tool to potentially define new operational paradigms for chemically reactive systems (deMello, 2006). One application of particular interest is the microfluidic separation or focusing of particles or droplets dispersed in a liquid. Such controlled particle migration can be accomplished in many different ways: via diffusophoresis (particle motion induced by a concentration gradient) (Fan et al., 2022; Shin, 2020; Liu and Pahlavan, 2025), dielectrophoresis (particle motion induced by a non-uniform electric field) (Çetin and Li, 2011; Midelet et al., 2019), magnetophoresis (particle motion induced by a magnetic field) (Forbes and Forry, 2012), magnetic biotransport (using magnetic particles as carrier particles to transport biomaterials) (Furlani, 2010), gravitational settling (leveraging gravitational effects to control particle trajectories) (Ma et al., 2011), or surface-pattern-based control of microparticles in the design of the microfluidic system (Hale and Mitchell, 2001). All of these variants require precise control of the behavior of the system to enable optimal and functional designs. The main challenge is to devise a microfluidic system in which particles suspended in a carrier fluid can be directed to a specific location with high accuracy and selectivity.

Despite the wide range of phenomena involved, all aforementioned methods are essentially based on the combination of Brownian motion, a non-conservative hydrodynamic force, and a conservative body force arising due to the application of an external field. The mathematical basis for deriving models on which to base numerical simulations is therefore very similar for microfluidic systems based on these principles. However, the situation is complicated by the fact that industrially relevant particle systems contain close to neutrally buoyant

[*] Corresponding author.
*E-mail address:* anandj@chalmers.se (A.J. Michael).

https://doi.org/10.1016/j.ijmultiphaseflow.2025.105421
Received 10 April 2025; Received in revised form 30 July 2025; Accepted 20 August 2025
Available online 28 August 2025

## Nomenclature

| | |
|---|---|
| $\Delta t$ | Time step size |
| $\gamma$ | Drag coefficient of spherical particle |
| $\boldsymbol{\sigma}$ | Viscous stress from fluid |
| $\mathbf{a}$ | Acceleration of particle |
| $\mathbf{F_B}$ | Brownian force on particle |
| $\mathbf{F_C}$ | Conservative force on particle |
| $\mathbf{F_H}$ | Hydrodynamic force on particle |
| $\mathbf{f}$ | Fluid body force |
| $\mathbf{K}$ | Hydrodynamic memory kernel tensor |
| $\mathbf{n}$ | Normal to particle surface |
| $\mathbf{u}$ | Fluid velocity |
| $\mathbf{V}$ | Velocity of particle |
| $\mathbf{x}$ | Position in domain |
| $\mu$ | Fluid viscosity |
| $\rho_f$ | Density of fluid |
| $\rho_p$ | Density of a particle |
| $\tau$ | Lag time into the past from current time |
| $\tau_p$ | Particle response time |
| $C$ | Cost function |
| $C_V$ | Velocity autocorrelation function |
| $D$ | Diffusion coefficient of unhindered spherical particle |
| $g$ | Acceleration due to gravity |
| $h$ | Distance of particle from wall |
| $k_B$ | Boltzmann constant |
| $L$ | Number of data points used for finding average particle statistics |
| $M$ | Total number of terms in the discretized memory kernel |
| $m^*_{\parallel}$ | Effective mass of particle in wall-parallel direction |
| $m^*_{\perp}$ | Effective mass of particle in wall-normal direction |
| $m_e$ | Effective mass of unhindered particle |
| $m_f$ | Mass of fluid occupying the same volume as particle |
| $m_p$ | Mass of particle |
| $n_K$ | Total number of known kernels used in a simulation |
| $p$ | Fluid pressure |
| $R$ | Impact Radius |
| $r_p$ | Radius of spherical particle |
| $Re$ | Reynolds number |
| $S$ | Particle surface |
| $T$ | Absolute temperature |
| $t$ | Current time |
| $V_{RMS}$ | Root Mean Square velocity |
| $v_z$ | Velocity in $z$ direction in simulation |
| $x_{ip}$ | Impact position of particle on wall along $x$ direction |
| $x_{n_K}$ | Position of particle in simulation with $n_K$ kernels |
| $y_{ip}$ | Impact position of particle on wall along $y$ direction |
| $z$ | Position in $z$ direction in simulation |
| $z_{DNS}$ | Position in $z$ direction in DNS simulation |
| $z_{LPT}$ | Position in $z$ direction in LPT simulation |

particles of sub-micrometer size (such as proteins, DNA, or other macromolecules (Ruggeri et al., 2018; Haghparas et al., 2025; Lee et al., 2018; Kroslak et al., 2007; Choi et al., 2020); viruses, bacteria, or cells (Arefi et al., 2020; Arefi, 2021; Bonilla et al., 2007); polymers (Mustin and Stoeber, 2010; De Santo et al., 2014); nanoparticles (Midelet et al., 2019; Unni and Yang, 2005, 2006, 2009; Cejas et al., 2019; Arefi et al., 2020; Cejas et al., 2017; Qiu et al., 2023; Sznitman, 2022); or asphaltene nanoaggregates (Lin et al., 2019; Mohammadghasemi et al., 2024)). Not only are such particles subjected to Brownian motion, which introduces a stochastic component to their trajectories in the microfluidic system (and hence make them more difficult to control (De Santo et al., 2014)), but the low particle-to-fluid density ratio also implies the action of significant nonlocal effects on particle migration. More specifically, the hydrodynamic force on the particles depends on the complete history of the development and rearrangements of the hydrodynamic boundary layers in the system (Daitche, 2015; van Hinsberg et al., 2011). These nonlocal effects also carry over to the Brownian motion via the fluctuation–dissipation relation (Hauge and Martin-Löf, 1973). At the same time, inertial effects are negligible, such that the lift forces are not important (Michaelides, 2016). Instead, hindrance effects arise when the particles interact with the walls of the microfluidic system. These hindrance effects modulate the hydrodynamic force and, for complicated system configurations, invalidate the traditional approach of approximating the hydrodynamic force as the steady drag force on a particle in an infinite reservoir (Kannan et al., 2019). Hence, simplistic models that describe particle motion as a balance between a simplified hydrodynamic force and a conservative force must be expected to fail.

Microfluidic devices for the aforementioned particulate types are typically associated with two important engineering challenges. When the microfluidic reactor acts as a sensor, where one would like to optimize the detection rate, detection level, and sensitivity (Choi et al., 2020; De Santo et al., 2014; Ruggeri et al., 2018). This implies that one wants to efficiently direct certain types of particles to a narrowly focused region somewhere within the device. At the same time, in most microfluidics applications, deterioration because of unwanted fouling is a critical issue due to the short distances to walls due to the miniaturization (Mohammadghasemi et al., 2024; Cejas et al., 2019; Yan et al., 2025; Lee et al., 2018; Lin et al., 2019; Mustin and Stoeber, 2010). Both of these challenges require high-fidelity simulation methods to enable *in silico* design and optimization.

Molecular simulation techniques (e.g. Brownian dynamics (Unni and Yang, 2005, 2006; Bonilla et al., 2007)) and continuum-based multiphase DNS techniques (Kannan et al., 2021; Michael et al., 2025) represent accurate approaches to modeling and simulation of hindered Brownian particle migration and deposition in microfluidic systems. However, as such simulations either describe the interactions among solute molecules at the particle level (Unni and Yang, 2005) or resolve the entire particle–fluid interaction (Michael et al., 2025), they remain computationally expensive. The high accuracy attainable with multiphase DNS, for example, allows detailed investigations of specific phenomena, but the computational cost associated with these techniques renders efficient collection of system statistics challenging. The development of new reactor designs for microfluidic particulate systems would benefit significantly from the establishment of more computationally efficient simulation methods, as long as their accuracy can be maintained close to that of DNS. As analytical models for hindrance effects for complex systems and particle configurations are not generally available, a universally applicable Lagrangian particle tracking (LPT) approach using a particle equation of motion with distinct force contributions cannot be established (Lee et al., 2018; Michaelides, 2016). Consequently, previous studies are primarily based on experimental investigations, supplemented by more general theoretical analyses.

In this work, we instead propose a novel nonlocal multiscale model for Brownian particles. This model combines an LPT routine with a multiphase DNS method to determine the hydrodynamic memory kernel of the particles. In this multiscale setup, the memory kernel (Hauge and Martin-Löf, 1973; Jung et al., 2017, 2018) can be optimally determined from short DNS sequences (Michael et al., 2025), providing DNS quality predictions of the nonlocal effects on hydrodynamic as well as Brownian forces. Particle trajectories can then be evolved over long times, and with variations in the random number sequence to collect proper statistics, in a Lagrangian frame of reference, which is much more computationally efficient and allows the investigation of realistic system sizes. The proposed model does not in any way depend on the availability of analytical expressions or correlations on how the hydrodynamic force varies due to interactions between particles and other nearby surfaces (Michaelides, 2016). There is also no need to exclude any contribution to the total hydrodynamic force by choosing a force model *a priori*. The only deterioration in accuracy as compared to a full DNS is related to the coarsened spatial resolution of the variation of the memory kernels with system configuration, the effect of which is systematically investigated and quantified.

The proposed model is used to investigate hindered deposition of particles in a microfluidic system in a channel flow configuration. In particular, we calculate the average area on the wall of a microfluidic system that a diffusing Brownian particle will strike, characterized by the average radius of the particle impact region referred to in this work as the impact radius. This is a key performance index in many microfluidic devices where particles need to be separated from or focused towards specific areas of the microfluidic device (De Santo et al., 2014). We show that there is a non-trivial variation of the impact radius with domain size and the strength of the conservative force attracting the particle to the wall. These results indicate that engineers designing microfluidic particulate systems can balance two independent design variables to tailor a specific system to the task at hand, thus illustrating the usefulness of accurate simulation models for the improved understanding and optimization of microfluidic systems.

## 2. Methodology

In this section, we introduce the novel nonlocal multiscale model developed to simulate particle migration in microfluidic devices. The multiscale framework has two constituent parts: a multiphase DNS component for the determination of particle memory kernels, and an LPT routine for evolving particle trajectories over long times.

As will become clear in the following, the new method can be used in two different ways: either a memory kernel library is pre-computed and used in subsequent Lagrangian simulations, or the memory kernels are computed on-the-fly as required by the Lagrangian framework. As the first approach is consistent with possible future developments, in which a machine-learning approach to determination of memory kernels based on a limited dataset of pre-computed kernels is used to supplement the model as presented here, we shall describe the model in accordance with the first way (a memory kernel library is pre-computed).

The method used to develop the memory kernel library is thus first described. The subsequent section details the implementation of the LPT method that utilizes the memory kernel library to simulate the particle motion. The final section of the methodology describes the sample cases set up to study how a particle migrates towards a wall in a microfluidic device using the developed methodology.

### 2.1. Determining memory kernels

The hydrodynamic force ($\mathbf{F_H}(t)$) acting on a particle immersed in a fluid can mathematically be modeled as the convolution integral between a hydrodynamic memory kernel tensor ($\mathbf{K}(t,\tau)$) and the velocity ($\mathbf{V}(t)$) history of the particle at low Reynolds numbers (Re $\ll 1$) (Hauge and Martin-Löf, 1973; Díaz, 2021):

$$F_{H,i}(t) = -\int_0^t K_{ij}(t,\tau)V_j(t-\tau)d\tau. \tag{1}$$

In case the particle is immersed in a fluid that is not stagnant, the relative velocity of the particle with respect to the fluid if it occupied the same position is used in place of the actual velocity of the particle.

The memory kernel is thus used to match the velocity history of the particle with the hydrodynamic force acting on the particle, allowing for the force to be nonlocal as the velocities of past times and previous locations are allowed to influence the current force. The hydrodynamic force in a particular direction, as shown in Eq. (1), is influenced by the history of the particle motion in all three orthogonal directions. The memory kernel tensor ($\mathbf{K}(t,\tau)$) at each lag time $\tau$ is a $3 \times 3$ tensor that represents how the velocity in different directions at time $\tau$ in the past from the current instance $t$ influences the force in a particular direction. Since the hydrodynamic force acting on the particle depends on the surroundings of the particle as well as the history of motion of the particle, the memory kernel contains information regarding the surroundings and can thus be thought of as a ”fingerprint” of the particle neighborhood and history of motion. This mathematical understanding of the hydrodynamic force has been used to generate memory kernel libraries in this work using optimization techniques, that is, by finding the memory kernel that optimally satisfies Eq. (1) for the available multiphase DNS data.

For a spherical particle, the memory kernel has been proven to be symmetric (Hauge and Martin-Löf, 1973). In the case of anisotropic particle or domain geometry, the non-diagonal terms of the memory kernel can become significant leading to cross-correlations between different directions. However, in the cases simulated in this work, the symmetrical nature of the domain and the spherical shape of the particles cause the non-diagonal terms of the memory kernel tensor to be negligibly small (Lauga and Squires, 2005; Simha et al., 2018). In order to reduce the numerical noise in the optimization procedure, it is then beneficial to simplify the memory kernel so that $K_{ij} = \delta_{ij}K_{ii}$ (Michael et al., 2025). In the case of an asymmetric particle or domain, the optimization techniques can be modified to include significant non-diagonal terms where required (Michael et al., 2025). Under the assumption that the particle does not move a long distance in a single time step, the memory kernel becomes independent of the time instance $t$ at which it is estimated, i.e. $\mathbf{K}(t,\tau) = \mathbf{K}(\tau)$. The hydrodynamic force then becomes:

$$F_{H,i}(t) = -\int_0^t K_i(\tau)V_i(t-\tau)d\tau. \tag{2}$$

It is then sufficient to determine the memory kernel in each direction corresponding to the hydrodynamic force in the same direction (conventional splits also used in this study comprise the normal (⊥) and parallel (‖) directions to the boundaries used in the simulations).

In this work, DNS data of a particle moving under the influence of a steady force is used to establish the memory kernels. These simulations are carried out using the mirroring Immersed Boundary (IB) method (Mark and van Wachem, 2008) incorporating Fluid–Structure Interactions (IB-FSI) implemented using in-house software (Mark et al., 2011). The mirroring IB method discretizes and solves the Navier–Stokes equation shown below to obtain the pressure and velocity fields of the fluid surrounding the particle:

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{3}$$

$$\rho_f\left(\frac{\partial u_i}{\partial t} + u_j\frac{\partial u_i}{\partial x_j}\right) = -\frac{\partial p}{\partial x_i} + \mu\frac{\partial^2 u_i}{\partial x_j^2} + f_i, \tag{4}$$

where $u$ is the fluid velocity, $x$ is the position in the domains, $p$ is the pressure, $\mu$ is the fluid viscosity and $f$ is any body force.

At the particle–fluid boundaries, an implicit Dirichlet IB boundary condition is applied such that the fluid velocity at the boundary is equal to the particle velocity at the same point. In practice, this is accomplished by mirroring the fluid velocity across the boundary of the particle such that linear interpolation between the velocity of the internal point and the external point in the fluid gives the boundary velocity. The pressure and velocity fields obtained from solving the Navier–Stokes equations using this method are then used to calculate the hydrodynamic force on the surface of the particle as shown in Eq. (5). Here, the pressure ($p$) and the viscous stresses ($\sigma$) are integrated over the surface ($S$) of the particle to give the total force exerted by the fluid on the particle.

$$\begin{aligned}\mathbf{F_H}(t) &= \int_S (-p\delta_{ij} + \sigma_{ij}) n_j \, dS, \\ &= \int_S \left( -p\delta_{ij} + \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right) n_j \, dS.\end{aligned} \tag{5}$$

The hydrodynamic force calculated in this manner is equivalent to the force modeled using the convolution integral between the memory kernel and the velocity history of the particle as shown in Refs. (Bian et al., 2016; Jaganathan et al., 2023):

$$\mathbf{F_H} = \int_S (-p\delta_{ij} + \sigma_{ij}) n_j \, dS = -\int_0^t \mathbf{K}(t,\tau) \cdot \mathbf{V}(t-\tau) ds \tag{6}$$

Thus, knowledge of the hydrodynamic force and the velocity history of the particle from the DNS method can be used to determine the memory kernel. To accomplish this, the particle is moved in the IB-FSI framework simulations by applying a constant force proportional to the variance of the Brownian force (Ounis et al., 1991; Li and Ahmadi, 1992; Kannan et al., 2019) acting on a particle of the same size immersed in the same fluid:

$$F_s = \sqrt{\frac{2k_B T\gamma}{\Delta t}}. \tag{7}$$

Here, $k_B$ is the Boltzmann constant, $T$ is the absolute temperature of the system, and $\Delta t$ is the size of the time step used in the simulation. The term $\gamma = 6\pi\mu r_p$, where $r_p$ is the particle radius, is the hydrodynamic drag coefficient for a spherical particle and represents a measure of the hydrodynamic resistance that an unhindered spherical particle experiences while moving in a fluid. The hydrodynamic force acting on the particle in response to the resulting particle motion and the corresponding particle velocity are recorded at each time step of the IB-FSI simulation.

Once the hydrodynamic force at a time instance and the velocity history of a particle leading to the same instance are known from the IB-FSI simulation, optimization techniques are applied to Eq. (1) to obtain the hydrodynamic memory kernel using the limited-memory Broyden–Fletcher–Goldfarb–Shanno optimization routine (L-BFGS-B) (Liu and Nocedal, 1989; Byrd et al., 1995; Zhu et al., 1997) from the `scipy` python library (Virtanen et al., 2020).

The cost function ($C(K_i(t,\tau))$) for the optimization routine is created from Eq. (2) by summing the hydrodynamic force acting on the particle with the convolution integral of the memory kernel and the velocity history, and normalizing the whole quantity:

$$C(K_i(t,\tau)) = \frac{|F_{H,i}(t) + \int_0^t K_i(t,\tau) V_i(t-\tau) d\tau|}{|F_{H,i}(t)|}. \tag{8}$$

In practice, the normalized cost function must be discretized in time according to:

$$C(K_i(t,\tau)) = \frac{|F_{H,i}(t) + \sum_{m=0}^{M} K_i(t, m\Delta t) V_i(t - m\Delta t)\Delta t|}{|F_{H,i}(t)|}. \tag{9}$$

Here, $M$ denotes the number of terms in the kernel. The memory kernel in each direction is initialized with zero for all its terms, and the values are then optimized against the cost function at each time step. Before the simulation has crossed $M$ time steps, only a subset of the values from the memory kernel are optimized, corresponding to the number of times steps completed by the simulation. At each new time step, one additional value of the memory kernel is optimized, until the number of time steps in the simulation crosses $M$, at which point the whole memory kernel is optimized at each time step against the corresponding velocity history. The memory kernel values from the previous time step are used as the initial guesses for the optimization process in the current time step to allow for faster convergence. This memory kernel optimization routine is explained in more detail by Michael et al. (2025). For this work, the IB-FSI simulations used to optimize the memory kernel were run with a time step of $\Delta t = \tau_p/50$ for a total duration of $20\tau_p$. The particle response time $\tau_p = m_p/\gamma$, where $m_p$ is the mass of the particle, characterizes how fast a particle responds to external stimuli. The response time is used in this work to normalize temporal quantities. The memory kernel ($\mathbf{K}(\tau)$) has sufficient terms to record the history effects for $\tau \in [0, 10\tau_p]$

### 2.2. *Evolving particle trajectories*

The Generalized Langevin Equation (GLE) can be used to describe the motion of a particle immersed in a fluid undergoing Brownian motion:

$$\begin{aligned} m_p \frac{d\mathbf{V}(t)}{dt} &= \mathbf{F_H}(t) + \mathbf{F_B}(t) + \mathbf{F_C}(t) \\ &= -\int_0^t \mathbf{K}(t,\tau) \cdot \mathbf{V}(t-\tau) d\tau + \mathbf{F_B}(t) + \mathbf{F_C}(t). \end{aligned} \tag{10}$$

Here, the total force acting on a particle can be split into a dissipative hydrodynamic force ($\mathbf{F_H}$), a random force representing Brownian forcing ($\mathbf{F_B}$) and a net conservative force ($\mathbf{F_C}$). The hydrodynamic force, as discussed in the previous section, can be represented as the convolution integral between the hydrodynamic memory kernel and the corresponding velocity history.

The random Brownian force and the hydrodynamic memory kernel are related via the fluctuation–dissipation theorem when the fluid is at thermal equilibrium (Hauge and Martin-Löf, 1973):

$$\langle F_{B,i}(t-\tau) F_{B,j}(t) \rangle = k_B T K_{ij}(t,\tau). \tag{11}$$

The GLE and its associated fluctuation–dissipation theorem show that the random Brownian force has a non-Markovian colored nature, with the random forces separated by a lag time $\tau$ exhibiting a correlation proportional to the hydrodynamic memory kernel value at the same time lag. These memory effects that also affect the Brownian motion are especially relevant at low particle–fluid density ratios (Mainardi et al., 2010; Hauge and Martin-Löf, 1973).

The memory kernel changes as a function of the hydrodynamic neighborhood of the particle. Thus, given a particular wall-bounded domain, the memory kernel associated with a particle will change depending on the particle position. For a simulation that causes the particle to traverse over wide ranges of the domain with different features, it is essential to figure out the memory kernel values at different positions along the trajectory. While, ideally, the kernel values along all positions traversed in the path of the motion of the particle are required, in this work we instead choose to build a library of memory kernels obtained at strategic positions along the particle trajectory. The memory kernel library, containing the kernel values $\mathbf{K}(\mathbf{x},\tau)$ corresponding to each sampled position $\mathbf{x}$, is built by running short IB-FSI simulations of the particle as described in Section 2.1.

Once the memory kernel library is built, the kernels can be used to simulate particle motion by utilizing them in the GLE shown in Eq. (10). The hydrodynamic force is now calculated as the convolution integral between the memory kernel in each direction and the velocity history of the particle in the same direction as shown in Eq. (2). The colored Brownian random force is determined based on the fluctuation–dissipation relation, which, when modified using the previously stated simplification of assuming that the non-diagonal terms of the memory kernel are zero (i.e. $K_{ij} = \delta_{ij} K_{ii}$), reduces to:

$$\langle F_{B,i}(t-\tau) F_{B,j}(t) \rangle = k_B T K_{ii}(t,\tau)\delta_{ij}. \tag{12}$$

Based on this relationship between the covariance of the Brownian force and memory kernel, the memory kernel scaled with $k_B T$ is used to initially build the covariance matrix for the random Brownian force at each time step. Once the covariance matrix is available, it can be factorized into root matrices which when multiplied with a series of normal random numbers with zero mean and unit variance generate the required colored random Brownian forces (Michael et al., 2025). The conservative force can be any body force arising due to the response of the particle to some sort of field (electrostatic, magnetic, gravitational, concentration, etc.) and can be modeled using traditional mathematical models for the same. In our simulation cases, we have used a constant force to simulate the attraction of a particle towards a surface, as explained further in Section 2.3.

The discretized form of the GLE (Eq. (10)) becomes:

$$\begin{aligned} m_p \frac{dV_i(t)}{dt} &= F_{H,i}(t) + F_{B,i}(t) + F_{C,i}(t) \\ &= -\sum_{m=0}^{M} K_i(t, m\Delta t) V_i(t - m\Delta t)\Delta t + F_{B,i}(K(t)) + F_{C,i}(t). \end{aligned} \tag{13}$$

The hydrodynamic force as well as the colored random Brownian force, are now derived from the memory kernel associated with the particle at its current position. They contain nonlocal information due to their construction via the convolution integral, and the combined framework constitutes a multiscale model as the memory kernels are obtained from DNS data with the particle trajectories evolved in an LPT routine.

The memory kernel at the current position ($\mathbf{x}(t)$) of the particle is derived from the memory kernel library by linear interpolation based on the position of the particle:

$$\mathbf{K}(t,\tau) = \mathbf{K}(\mathbf{x_1},\tau) + \frac{(\mathbf{x}(t) - \mathbf{x_1})}{\mathbf{x_2} - \mathbf{x_1}}(\mathbf{K}(\mathbf{x_2},\tau) - \mathbf{K}(\mathbf{x_1},\tau)) \quad \text{where } \mathbf{x_1} < \mathbf{x}(t) < \mathbf{x_2}. \tag{14}$$

Thus, at time $t$ if the particle is in a position $\mathbf{x}(t)$ such that $\mathbf{x_1} < \mathbf{x}(t) < \mathbf{x_2}$, then the memory kernel associated with the particle $\mathbf{K}(t,\tau)$ can be obtained by linear interpolation between $\mathbf{K}(\mathbf{x_1},\tau)$ and $\mathbf{K}(\mathbf{x_2},\tau)$. Here, $\mathbf{x_1}$ and $\mathbf{x_2}$ are the positions at which the memory kernel has been determined and stored in the kernel library.

The hydrodynamic force ($\mathbf{F_H}$) and the random Brownian force ($\mathbf{F_B}$) generated using the interpolated memory kernel are added to the conservative force ($\mathbf{F_C}$) to arrive to the net force acting on the particle. The acceleration the particle experiences can then be derived as shown:

$$\mathbf{a}(t) = \frac{\mathbf{F_H}(t) + \mathbf{F_B}(t) + \mathbf{F_C}(t)}{m_p}. \tag{15}$$

The velocity and position of the particle are then modified using the Newmark scheme (Newmark, 1962). The acceleration $\mathbf{a}(t + \Delta t)$ refers to the implicit acceleration at the new position reached, based on the time-stepping scheme. This implicit acceleration is iteratively derived by assuming that the force balance holds between the hydrodynamic and conservative force at the new position while maintaining the same Brownian force:

$$\mathbf{V}(t + \Delta t) = \mathbf{V}(t) + \frac{\Delta t}{2}\left[\mathbf{a}(t) + \mathbf{a}(t + \Delta t)\right], \tag{16}$$

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \mathbf{V}(t)\Delta t + \frac{\Delta t^2}{2}\left[\mathbf{a}(t)\right]. \tag{17}$$

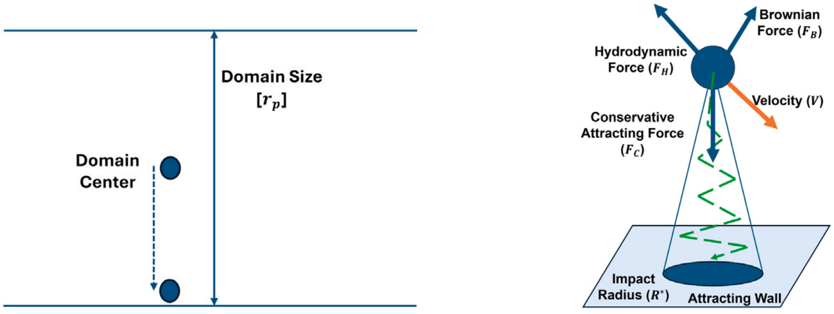
While explicit time stepping schemes are traditionally used to solve such Stochastic Differential Equations (SDEs), the Newmark implicit scheme has been implemented in this work to maintain consistency between the IB-FSI framework used to estimate the memory kernel and the LPT simulations, which then use these kernels in their implementation. The IB-FSI framework requires an implicit scheme for stable convergence of the solution while moving the particle as part of the DNS simulations. These can influence the values of the memory kernel, and hence a similar time stepping scheme in the LPT simulations ensures cross-framework consistency in the hydrodynamic force generation (Michael et al., 2025). It has previously been shown that the implicit Newmark scheme is able to capture the Mean Square Displacement (MSD) and the Velocity Autocorrelation Function (VACF) trends in both short and long terms (Kannan et al., 2019, 2021; Michael et al., 2025). The small time steps used in the simulations (i.e. $\Delta t = \tau_p/50$) are also selected to minimize the differences in individual paths between implicit and explicit versions of the simulation.

Thus, the simulation process starts with the determination of the memory kernels at different positions in the simulation domain using DNS based on the IB-FSI framework. A kernel library is created mapping the hydrodynamic behavior of the particle as it moves through the domain. The kernels from the library are utilized to move the particle in an LPT framework using the GLE. During the LPT simulation, interpolation is used to determine the memory kernel associated with the current particle position. Once the kernel is determined, the hydrodynamic force and the random Brownian force acting on the particle are created from the kernel based on the GLE and the fluctuation–dissipation relationship. The net force on the particle incorporating hydrodynamic, Brownian and conservative components are finally employed to propagate the particle.

### 2.3. Microfluidic cases

We employ the new memory-kernel based LPT methodology to investigate the attraction of target particles to the surface in wall-bounded microfluidic domains. The domain is considered unbounded in the $x$ and $y$ directions, while there are walls present at the top and the bottom in the $z$ direction. The distance between the walls is set to $5r_p$, $10r_p$ and $30r_p$, to study the effect that the variation in the wall distance and the geometrical extent of the reactor have on the particle dynamics. The large domain of size $30r_p$ is chosen because when the particle is placed in the center of the domain, it is not influenced by the wall. In contrast, as the domain size is reduced to $10r_p$ and $5r_p$, the particle experiences increasing wall effects even when placed in the center of the domain. Thus, the various sizes of the domain were chosen to show how the wall effect increasingly influences the particle dynamics and the shapes of the kernel as the domain size decreases. The particle enters the center of the domain between the walls and then approaches the bottom wall under the influence of a steady conservative force, while also undergoing Brownian motion. This attracting force is set to either $m_p g$ or $5m_p g$, where $m_p$ is the mass of the particle and $g$ is the gravitational constant, to demonstrate how the magnitude of the attracting force affects the particle deposition behavior in the two domains. A representation of the case domain and the starting position of the particle is shown in Fig. 1(a). A representation of how the particle moves following a meandering path under the influence of the hydrodynamic, Brownian and conservative attractive forces is shown in Fig. 1(b).

The objectives of the simulations are to capture the behavior of the particle in terms of position and velocity as it moves from the center of the domain towards the wall. In particular, the influence of the wall on the approach of the particle towards the wall under the application of the attractive force is studied using this information. Controlling the particle impacting location on the walls of a microfluidic device is also of interest, as mentioned in the introduction, so the final impact location on the bottom wall at the end of the simulations is also recorded. This indicates how the device geometry and the attracting force influence the diffusion of the Brownian particle in the $xy$ plane in the time it takes the particle to hit the wall. Since the particle in each iteration of the simulation can be captured at a different point on the wall, the average radius on the wall where the particle can impact is represented in this figure as the impact radius (R*), which will be described in further detail in Section 3.3.



(a)

(b)

**Fig. 1.** Figure (a) gives a general representation of the domain used in the settling cases. Figure (b) shows a representation of how the particle starting from the center progresses towards the bottom attracting wall while experiencing hydrodynamic, Brownian and conservative forces. The average radius within which the particle impacts the wall is represented here as the impact radius.

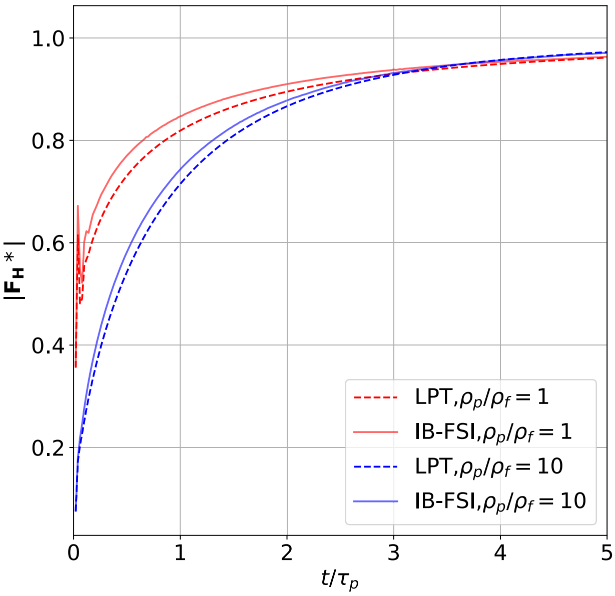


**Fig. 2.** The hydrodynamic force obtained using the memory-kernel-based LPT and the multiphase DNS using IB-FSI for a non-Brownian particle accelerated by a steady force.

The memory kernel libraries for all the domains used in this work are developed by running IB-FSI simulations of the particle at different $z$ positions. As shown in Fig. 1(a), the domains used in the determination of the memory kernels consist of two walls at the top and the bottom in the $z$ direction. In the $x$ and $y$ directions, the domains extend for $15r_p$ from the particle position to minimize any potential boundary influence on the kernel values. The simulations in the wall-bounded domains are run at a particle–fluid density ratio ($\rho_p/\rho_f$) of 1. For this density ratio, the added mass effect is the dominant force compared to the drag and history effects (Michaelides, 2016). Theoretical estimates of the effective mass (including the added mass) for a particle close to a wall take the following form (Simha et al., 2018):

$$m^*_\perp = m_p + \frac{m_f}{2}\left[1 + \frac{3}{8}\left(\frac{r_p}{h}\right)^3\right], \qquad (18)$$
$$m^*_\parallel = m_p + \frac{m_f}{2}\left[1 + \frac{3}{16}\left(\frac{r_p}{h}\right)^3\right].$$

The symbols ∥ and ⊥ indicate the added mass effect in the directions parallel to and normal to the wall, respectively. Here, $m_f = m_p\rho_f/\rho_p$ is the mass of the fluid if it occupied the same volume as the particle and $h$ is the distance of the particle center to the wall. It can be seen from these equations that the increase in the added mass as the particle moves closer to the wall is proportional to $(r_p/h)^3$. The positions in the $z$ direction used to create the memory kernel library were developed on the basis of this formula. The distances of the particle from the wall ($h$) were chosen starting from the center (for example $h = 15r_p$ in the domain of size $30r_p$) such that the term $(r_p/h)^3$ increased by the same amount as the particle was shifted closer to the wall. The positions used to sample and create the memory kernel library from the IB-FSI simulations are shown in Table 1 for each domain.

In the present work, we focus on the role of hydrodynamics and thermal fluctuations of the suspended particles, such that the computational framework fully accounts for particle–particle and particle–wall interactions mediated via the fluid fields. Van der Waals attraction and electrostatic repulsion, collectively referred to as Derjaguin–Landau–Verwey–Overbeek (DLVO) interactions (Cejas et al., 2019; Haghparas et al., 2025), which occur over very short distances, are not included. The distance at which DLVO interactions become significant to the hydrodynamic force on a particle approaching a wall depends on the particle velocity, as well as the properties of the wall–particle–fluid system. For a neutrally buoyant Brownian particle moving in water at moderate to high ionic strength towards a silicon wall and at a drift velocity resulting from the application of an external conservative force of the orders of magnitude considered in the current work, the DLVO interactions remain insignificant down to distances of approximately one percent of the particle diameter. This situation supports the decision to neglect these effects in the current analysis. If the particle volume fraction in the near-wall region becomes high, rheological and structural effects induced by electric double layer forces could possibly emerge (Srinivasan et al., 2022). The extension of the current method to also include van der Waals and electric double-layer effects as forces in the GLE (Eq. (10)) in such situations is however, straightforward, as the resulting force can be derived from potential energy functions and therefore is conservative by definition (Lyklema et al., 1999). Thus, a DLVO force can be combined with the present $F_C$ without further changes to the numerical methodology developed here, if the current methodology is to be applied to systems where these interactions are deemed important.

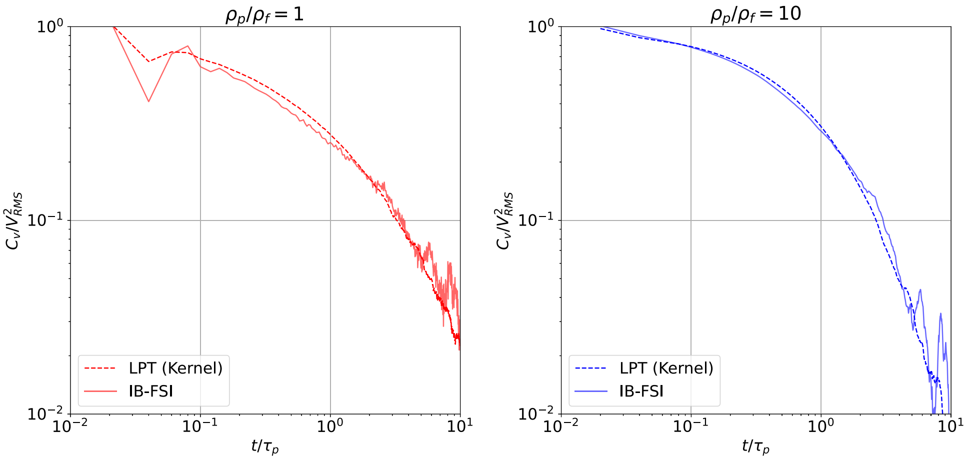


**Fig. 3.** Velocity Autocorrelation Function (VACF) comparison between the GLE-based DNS (IB-FSI) and GLE-based LPT using memory kernels from IB-FSI.

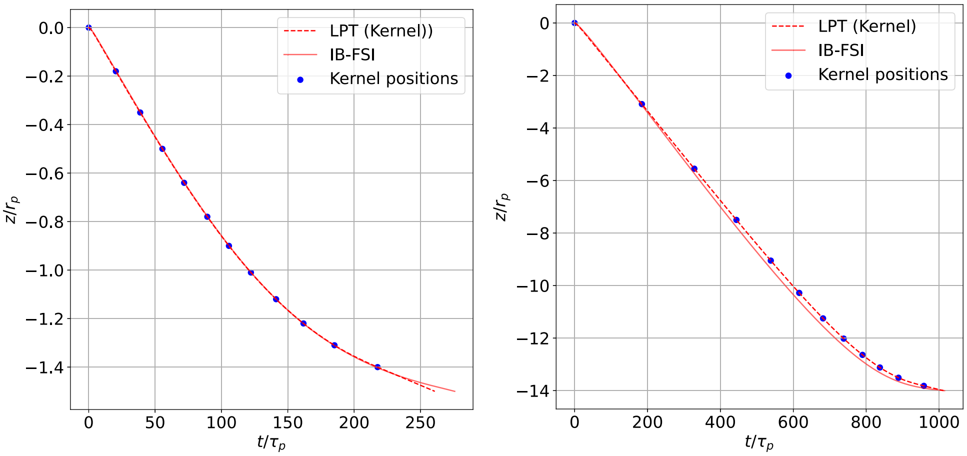


**Fig. 4.** Comparison of trajectories obtained from settling simulations run using DNS (IB-FSI) and the LPT method utilizing linear interpolation of hydrodynamic memory kernels. The figure on the left shows settling in the $5r_p$ domain while the figure on the right shows settling in the $30r_p$ domain. The positions at which the kernels are known from previous DNS simulations are shown as blue dots overlaid on the trajectories obtained from the LPT simulations.

**Table 1**
The particle positions represented as a distance from the wall ($h$) used for sampling the memory kernel in the IB-FSI simulations for the two different domain sizes.

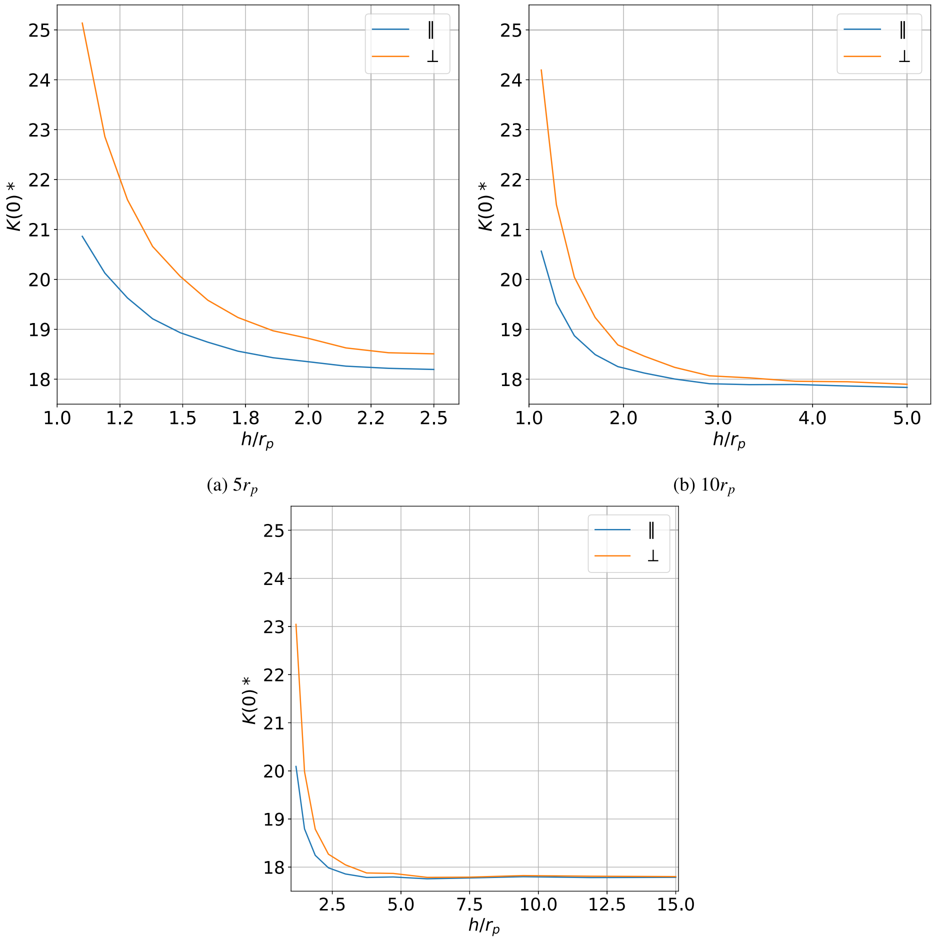
| Domain | Particle position ($\mathrm{h/r_p}$) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **30 $\mathbf{r_p}$** | 15.00 | 11.91 | 9.45 | 7.50 | 5.95 | 4.72 | 3.75 | 2.98 | 2.36 | 1.88 | 1.49 | 1.18 |
| **10 $\mathbf{r_p}$** | 5.00 | 4.37 | 3.82 | 3.33 | 2.91 | 2.54 | 2.22 | 1.94 | 1.70 | 1.48 | 1.29 | 1.13 |
| **5 $\mathbf{r_p}$** | 2.50 | 2.32 | 2.15 | 2.00 | 1.86 | 1.72 | 1.60 | 1.49 | 1.38 | 1.28 | 1.19 | 1.10 |

## 3. Results and discussion

### 3.1. Verification and validation

The novel nonlocal multiscale method is verified and validated by running a simulation where the particle is pushed using a steady force equivalent to that shown in Eq. (7) in the absence of Brownian motion. The hydrodynamic force ($\mathbf{F_H}$) modeled using the convolution integral between the memory kernel and the velocity history of the particle in this case is recorded. The equivalent simulation is also performed in full DNS (using the IB-FSI method) and the corresponding hydrodynamic force (obtained by carrying out the surface integral of the viscous stresses and pressure exerted by the fluid on the particle) is recorded here as well. The comparison between the hydrodynamic forces obtained in both these manners in the initial $5\tau_p$ of the simulation is presented in Fig. 2. Both the memory-kernel-based LPT simulation and the IB-FSI simulation are run at particle-to-fluid density ratios of 1 and 10 to demonstrate the validity of the method at varying density ratios. The steady force used to push the particle has been used to normalize the hydrodynamic force in all the cases (an asterisk denotes a normalized value, $\mathbf{F^*_H}$). It can be seen that the forces obtained from the memory-kernel-based LPT simulations closely follow the force produced in the IB-FSI simulations. There are minor differences in the initial few times steps, but in the long term they tend to the same force in both cases. The differences are somewhat more pronounced at the lower particle-to-fluid density ratio.

To verify that the simulations are capable of simulating Brownian motion accurately, the memory kernel was determined for an unhindered Brownian particle immersed in a fluid. The unhindered configuration implies that there are no other objects near the particle that could alter the hydrodynamic forces acting on the particle. The memory kernel determined was used to run an LPT simulation as described in Section 2.2. The same simulation is also carried out in the more computationally expensive but also more accurate IB-FSI



(a) $5r_p$

(b) $10r_p$

(c) $30r_p$

**Fig. 5.** Peaks of memory kernels ($K^*(0)$) in the domains with walls at distances of $5r_p$, $10r_p$ and $30r_p$.

framework, where the memory kernel is determined on-the-fly at each time step. To characterize the Brownian motion, the VACF (or $C_V$) for the particle is calculated for both simulations at different lag times:

$$C_V(\tau) = \langle V(0) \cdot V(\tau) \rangle = \frac{1}{L} \sum_{t=0}^{t=T-\tau} [V(t) \cdot V(t+\tau)] \,. \tag{19}$$

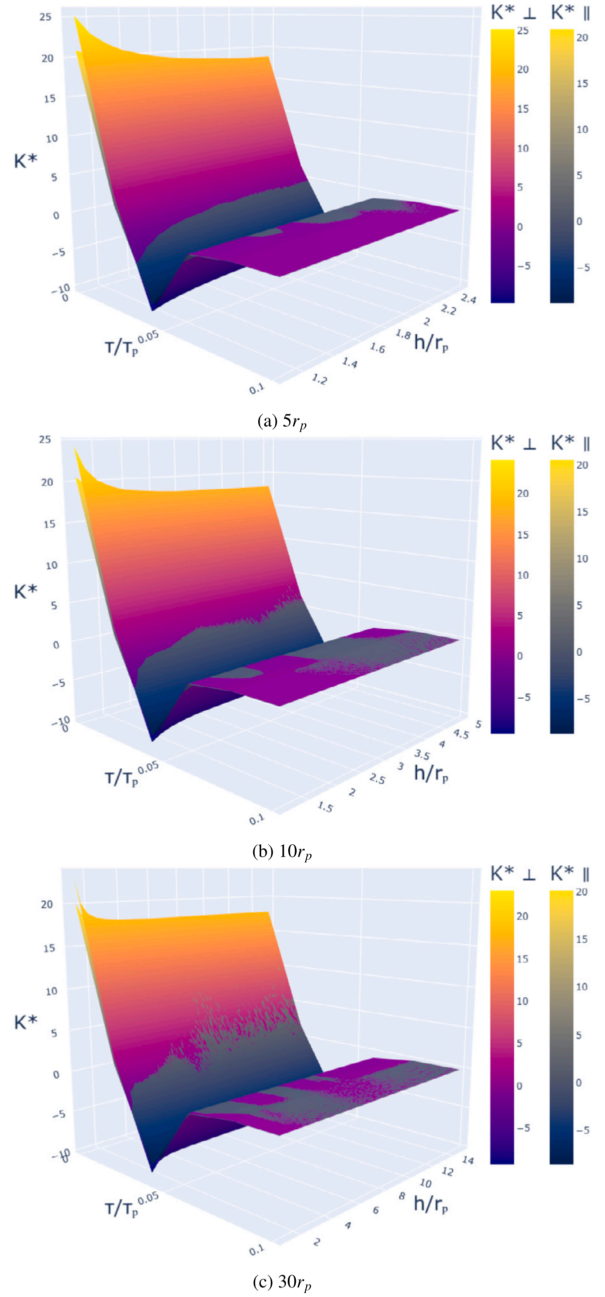
Here, $V(t)$ is the velocity of the Brownian particle at time $t$, $\tau$ is the time lag by which the two velocities used for the correlation are separated, and $L$ is the number of data points used to create the correlation. The $C_V$ calculated in this manner is normalized using the expected Root Mean Square velocity ($V_{RMS}$) for an unhindered particle at the given particle-to-fluid ratio. The $V_{RMS}$ velocity is given as $V_{RMS} = \sqrt{3k_B T/m_e}$, where $m_e = m_p(1 + \rho_f/2\rho_p)$ is the effective mass of the particle that includes the added mass effect from the fluid dragged along with the particle (Mainardi et al., 2010).

The normalized VACFs obtained from the memory-kernel-based LPT and IB-FSI frameworks are depicted in Fig. 3. The VACF lines obtained from the nonlocal multiscale LPT framework proposed here, for both particle-to-fluid density ratios of 1 and 10, can be seen to overlap well with the lines obtained from IB-FSI framework which was shown in the work by Michael et al. (2025) to accurately model Brownian motion. The oscillations towards the larger time lags are due to lower number of data points available for creating the correlations. The proximity of the hydrodynamic force lines (Fig. 2), as well as the VACF lines obtained from the LPT framework to those obtained from the IB-FSI framework (Fig. 3), show that the new method is capable of accurately modeling hydrodynamic effects while also incorporating Brownian motion.

As previously mentioned, for the positions between those specified in Table 1, the memory kernel is determined via linear interpolation between known kernels as shown in Eq. (14). To verify the accuracy of this approach, pure settling simulations are carried out in the domains, excluding any Brownian effects, using the IB-FSI framework, as well as the LPT framework. The settling force used in this case was set to $5m_p g$, to traverse the full position range, starting from the center of the domain to the wall. The trajectories obtained in both cases are shown in Fig. 4 for the smallest domain of $5r_p$, as well as for the largest domain of $30r_p$. We observe that the trajectories obtained using LPT simulations employing linear interpolation between the kernels are very close to those obtained from DNS using the IB-FSI framework. The blue marks on the trajectory indicate the positions of the known kernel values. At all other points along the trajectory, the kernels have been determined using linear interpolation. After the particle crosses the last position where the kernel is known, the kernel values are held constant until the particle is captured by the wall, leading to some deviation from this point forward from the DNS trajectory.

$$Error(z) = \frac{\sum_{i=0}^{n} |(z_{LPT,i+1} - z_{LPT,i}) - (z_{DNS,i+1} - z_{DNS,i})|}{z_{DNS,n}} \tag{20}$$

The error in the trajectory between the DNS and the LPT simulations is calculated using Eq. (20), where the numerator gives a measure of the relative difference in positions between the trajectories obtained from the LPT and DNS simulations, while the denominator gives the total distance traversed by the particle. The error up to the final position where the kernel is known is approximately 1% and 4.8% for the $5r_p$



(a) $5r_p$

(b) $10r_p$

(c) $30r_p$

**Fig. 6.** The initial parts of the 12 memory kernels ($K^* = K(h,\tau)\Delta\tau/2\gamma$) sampled from the bounded domains are shown as surface plots against time ($t$) and the distance of the sampling point from the wall ($h$). There are two surfaces plotted in each figure corresponding to the kernel associated with the directions normal (⊥) and parallel (‖) to the wall. As the particle moves closer to the wall, the associated peaks and valleys in the initial part of the memory kernel attain higher values, indicating the increased drag and added mass effects that the particle experiences close to the wall. The increase is more pronounced in the wall-normal direction as compared to that in the parallel direction. Away from the wall, the kernels converge to the same values irrespective of the direction indicating the decreasing influence from the wall.

and the $30r_p$ domain, respectively. The error is more pronounced in the larger domain due to the sparser sampling of kernels overall. But apart from that, in the center of the domain, the history effects die out much more slowly, meaning that longer kernels are required for a more accurate representation of the hydrodynamics (Michael et al., 2025). However, as previously mentioned, the kernel lengths have been limited in the simulations to reduce computational cost, leading to some minor loss of hydrodynamic information in unhindered scenarios. In the near-wall regions, where the history effect decays much faster, the tail of the kernel correspondingly decays faster, contributing less to the force model. Therefore, shorter kernels could be used in the smaller domains to get accurate trajectories, since wall effects will dominate in such domains. Beyond the final point where the kernel is known, the deviation from the DNS path caused by maintaining the kernel constant is 2.3% and 0.7% for the $5r_p$ and the $30r_p$ domain, respectively. The increased hindrance effect (drag and added mass) in the smaller domain likely causes a larger deviation after the final sampling point when compared to the larger domain. In cases where the wall adjacent region is of primary interest, denser sampling of the kernels in this area can minimize such deviations. The LPT simulations utilizing the linear interpolation between the kernels are thus able to capture the DNS results quite accurately, thereby paving the way for the settling simulations incorporating Brownian motion as well.

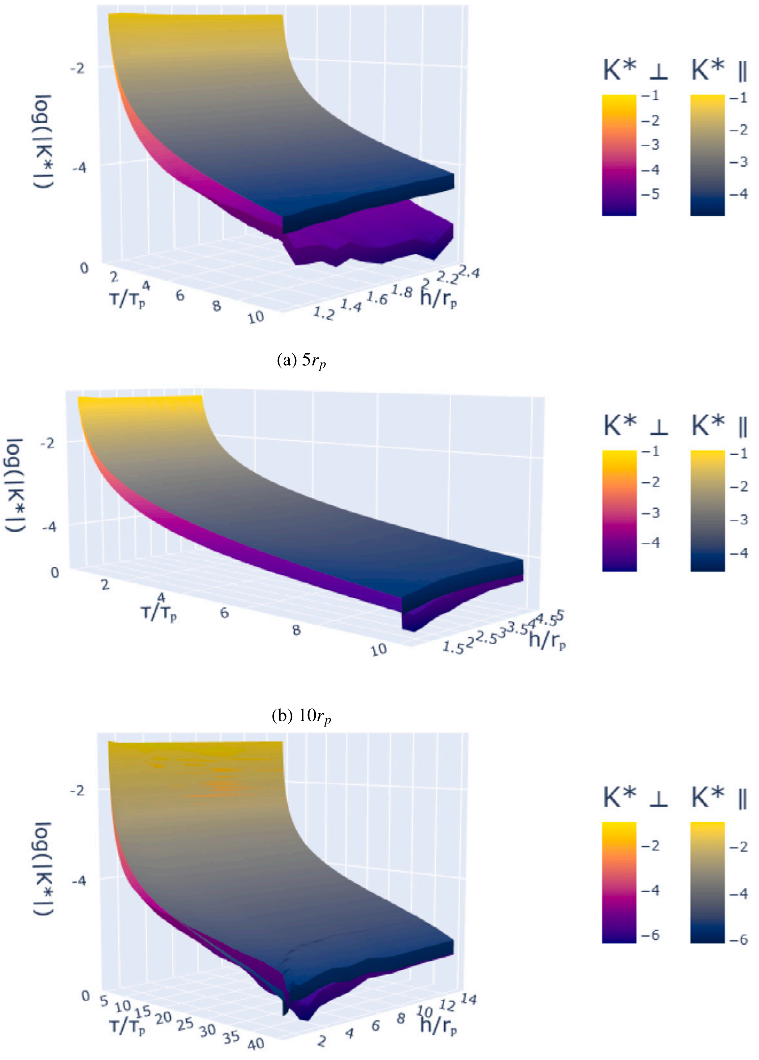
### 3.2. Characteristics of the memory kernels

The nonlocal multiscale method is used to simulate hindered particle deposition on a target wall in wall-bounded domains where the walls are separated by a distance of either $5r_p$, $10r_p$ or $30r_p$. The memory kernels in these cases were obtained by running brief IB-FSI simulations at the locations shown in Table 1. These simulations were run at a particle-to-fluid density ratio of 1 (i.e. neutrally buoyant), which is most relevant to the microfluidic particulate systems of interest.

The peaks of the memory kernels (that is, their value at $K(\tau = 0)$) in the domains are shown in Fig. 5. To provide a more comprehensive illustration of the richness of data contained in the memory kernel library, the initial parts of the memory kernel, as well as the tails, are also illustrated (Figs. 6 and Fig. 7 respectively). The peaks are predominantly an indication of the magnitude of the drag and the added mass effect, while the tail of the kernel depicts how the history effect decays with time. It can be seen in Figs. 5 and 6 that, as the particle moves closer to the wall, the peaks of the kernels increase, indicating that the drag and the added mass effects increase, as expected from theoretical predictions (Felderhof, 2005). The changes in the peaks and tails of the kernels are more pronounced in the normal direction to the wall than in the parallel direction to the wall, as seen from the wider separation between the kernel values in each direction in Figs. 6 and 7, indicating the stronger influence of the wall on the dynamics of the particles in the normal direction. The memory kernel inherently carries information regarding the hydrodynamic effects that the presence of the wall will have on particle dynamics. The memory kernel, thus, removes the need for analytical mathematical models that are otherwise required to carry out LPT simulations, incorporating the various hydrodynamic effects observed at such a low particle-to-fluid density ratio, along with their modulation due to the presence of walls and other particles.

### 3.3. Particle migration in microfluidic systems

In this section, we use the method developed herein to analyze the particle segregation from a fluid in microfluidic systems of varying geometrical size. We do this by also varying the magnitude of the steady force that attracts the particles to the target wall. The layout of the systems analyzed was presented in Section 2.3. The particle starts from rest at the centerline between the bounding walls of each domain, and moves towards the bottom wall of the domain under the influence of a constant attractive force while also undergoing Brownian motion in all directions. The simulations are run until the particle deposits on the target wall, i.e. when the particle center is at a distance of a particle radius from the attracting wall. The random numbers required for the generation of the Brownian force in these simulations were generated by using the `random` module in Python (Python Software Foundation, 2025). For each domain and force combination,



(a) $5r_p$

(b) $10r_p$

(c) $30r_p$

**Fig. 7.** The logarithmic values of the tails of the 12 memory kernels ($K^* = K(h,\tau)\Delta\tau/2\gamma$) sampled from the bounded domains are shown as surface plots vs. time ($t$) and the distance of the sampling point from the wall ($h$). There are two surfaces plotted in each figure corresponding to the kernel associated with the directions normal (⊥) and parallel (∥) to the wall. The dip in the tail indicates the rate at which the history effect decays. The rate of decay is more pronounced in the wall-normal direction as compared to that in the parallel directions, indicating that the force decorrelation is stronger in this direction. Away from the wall, the tails converge to a common surface irrespective of orientation, indicating the decreasing influence from the wall.

10 random number seeds were used to create 10 variations of the same simulation for ensemble averaging of the statistics. To understand the effects of the wall on the particle dynamics, unhindered simulations are also performed where the particle is allowed to migrate over the same distance as in the corresponding wall-bound case, but with the assumption that the wall is far enough from the particle to not affect its motion. This section focuses on comparing the hindered simulations based on domain size and forces utilized, while the next section focuses on a comparison between the unhindered and hindered simulations.

Since there are no external forces in the $x$ and $y$ directions, the particle undergoes pure Brownian motion in these directions. In contrast, in the $z$ direction, the steady force causes the particle to migrate towards the wall. The position of the particle in the $z$ direction is depicted as a function of time in Fig. 8. The velocity of the particle in the $z$ direction is presented against the particle position in the same direction in Fig. 9. In these figures, time is normalized with the particle response time ($\tau_p$), position is normalized with the particle radius ($r_p$), and particle velocity is normalized with the RMS velocity ($V_{RMS}$).

In Fig. 8, the green lines indicate the trajectories of the particle at the lower force of $m_p g$ while the red lines indicate the trajectories at the larger force of $5m_p g$. The dashed lines in magenta and blue indicate the unhindered trajectories at the force of $m_p g$ and $5m_p g$, respectively. The endpoints of the trajectories in each domain are indicated by the correspondingly labeled horizontal lines. It can be seen that the slope of the position graph is steeper for the simulations performed with the higher conservative force. The slope is initially almost identical in the $10r_p$ and $30r_p$ domains, indicating that there are no significant effects from the presence of the wall on the channel centerline. In contrast, the $5r_p$ domain exhibits a lower slope from the very beginning, indicating that wall effects are present even in the center of the domain in this case. In all cases, as the particle approaches the wall in each domain, it begins to slow down, as seen from the reduced slopes in the position graph.

Similarly, in Fig. 9, the velocity value is plotted against the position of the particle in the domain. Here, labeled vertical lines are used to indicate the end of the trajectory in each domain. The particle velocity can be seen to approach higher values initially in the simulations in which the particle experiences a higher conservative force. The particle initially attains a terminal velocity that is proportional to the force applied to direct it towards the wall. This velocity, however, changes

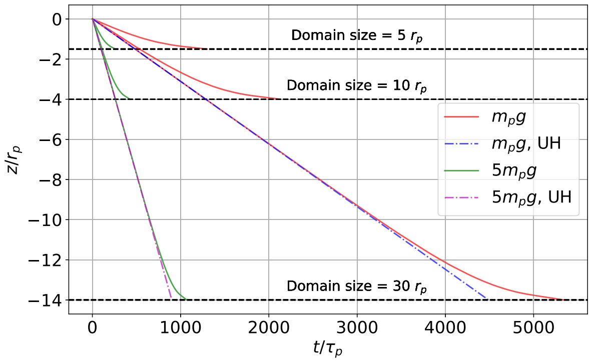


**Fig. 8.** The position of a particle in the $z$ direction as the particle migrates towards the bottom wall is depicted against time. A force equivalent to either $m_pg$ or $5m_pg$ is used in each of the reactor geometries. The label UH is used here to identify the unhindered simulations.

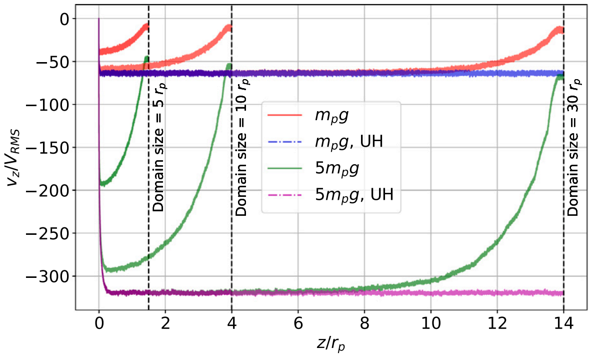


**Fig. 9.** The change in the particle velocity in the $z$ direction is plotted against the particle position along the same direction. A force equivalent to either $m_pg$ or $5m_pg$ is used in each of the reactor geometries. The label UH is used here to identify the unhindered simulations.

to lower values as the particle approaches the wall and hindrance effects start to manifest. The particles seem to approach a new terminal velocity close to the wall at the end of their trajectory. However, this is because, for every position after the final sampling position used to create the memory kernel library (cf. Table 1), the particle uses the memory kernel associated with the final sampling point to simulate the particle motion (no extrapolation). This induces some inaccuracies in the final approach towards the wall, which can be avoided by increasing the frequency of kernel sampling close to the wall or by opening up for extrapolation of the memory kernel values outside the library domain. As seen from the settling simulations used to verify the validity of the model in Section 3.1, the errors due to this lack of extrapolation beyond the final kernel sampling are quite small.

The most important measure of the degree of control attainable for a given microfluidic system design, the impact position in the $xy$-plane, is recorded for each particle trajectory completed. The radius of impact is calculated as $R = \sqrt{x_{ip}^2 + y_{ip}^2}$, where $x_{ip}$ and $y_{ip}$ are the final impact positions in the $x$ and $y$ directions respectively. The impact radius is then normalized to $R^* = R/\sqrt{4D\tau_P}$, where $D = k_bT/\gamma$ is the diffusion coefficient for an unhindered spherical particle and $\sqrt{4D\tau_p}$ is a measure of the distance that an unhindered particle would diffuse over in a plane during the time $\tau_p$. The normalized impact radius is thus a measure of how far the particle has diffused in the $xy$-plane by the time that the particle approaches the wall. Diffusion realistically depends on the time available for diffusion as well as the hydrodynamic hindrance experienced by the particle. As can be seen from the formula for the diffusion coefficient, the diffusion is inversely proportional to the hydrodynamic hindrance, and thus, the larger the hindrance, the lower the corresponding diffusion (Felderhof, 2005).

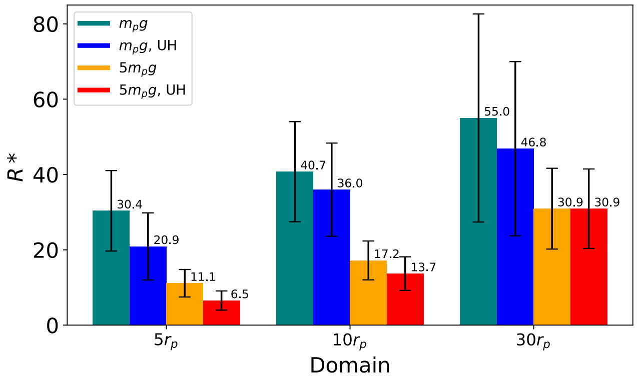


**Fig. 10.** The normalized impact radius ($R^*$) for the domains at two magnitudes of the attracting force. The bars represent the average impact radius and the error bars represent the variance in the impact radius, as obtained from ensemble-averaging over repeated particle trajectory evolutions. The value of the average impact radius in each case is indicated on top of the bars.

In Fig. 10, the average impact radius for each domain and force combination is shown for the hindered as well as unhindered scenario. The impact radii were grouped by domain and labeled according to the forces used in each case. While the bars indicate the average impact radii, the error bars illustrate the variance of the impact radius. The green bars indicate the impact radii when the force is $m_pg$ and the orange bars indicate the same when the force is $5m_pg$ in the hindered cases. It can be seen that the impact radius, as well as its variance, reduces as the force applied to the particle increases in the same domain. Since the particle spends less time in the domain in the higher-force scenario (it reaches the target wall sooner), it has less time to diffuse, and this is observed as the reduced impact radius. For the same conservative force, when the domain size is increased, the impact radius also increases as the particle spends more time diffusing parallel to the wall in a larger domain. As was seen from the evolution of particle positions and velocities, the hindrance to particle motion increases drastically only as the particle enters the near-wall region. Therefore, in larger domains, where the Brownian particle spends more time in the low-hindrance area away from the wall, it is to be expected that the particle will show a larger impact radius when it eventually arrives at the wall.

It can be seen from Fig. 10 that, within the parameter space investigated in the current work, the largest impact radius (55) is observed in the $30r_p$ domain at the force of $m_pg$, while the smallest impact radius (11.1) is observed in the $5r_p$ domain at the force of $5m_pg$. Based on the difference between these values, it can be seen that the optimal design of a microfluidic device can reduce the impact radius by almost 80%. More specifically, this is achieved by amplifying the hindrance (making the device smaller or confining the geometry along the path to the focus point on the target wall) while increasing the magnitude of the attracting force (increasing the strength of interaction between the external field and the particle or modulating the particle response to the same field). These observations illustrate how insights from simulations can help untangle the confounded effect of parameters through which the process may be optimally designed and controlled.

### 3.4. The role of hindrance in the deposition process of particles

To better understand the effects of the wall on the approach of the particle to the target area in the microfluidic reactor, and also the role of the hindrance arising from the presence of the wall on the possibilities to focus the impact radius, a comparison between unhindered and hindered Brownian motion is performed. In the unhindered simulation, the Brownian particle is pulled using a steady force towards the wall, but here the wall is considered to be far enough to not have an influence on the particle motion. The forces and domains used are the same as those described in the previous section. Here, the main difference from the hindered scenario described in the previous section is that only the hydrodynamic kernel associated with a particle located at the center of the largest fluid domain is used to move the particle. This is implemented by using the kernel associated with the center of the $30r_p$ domain. Since the particle is at a significant distance from the wall, the hydrodynamic force, as well as the random Brownian force modeled using the memory kernel associated with this position, are considered equal to those of an unhindered particle (i.e. the influence of the wall is considered negligible). Similarly to the previous cases, 10 different random number seeds have been used in the unhindered settling cases to create enough data to elucidate the statistical tendencies.

In Fig. 8, the dashed blue and magenta lines indicate the trajectory of the settling Brownian particle in the $z$ direction in the unhindered case for one realization of the simulation at the forces of $m_pg$ and $5m_pg$, respectively. The trajectories collapse onto the same line irrespective of the domain size traversed by the particle due to the lack of any wall effects. Similarly, in Fig. 9, the corresponding velocity of the same migrating particle in the $z$ direction is depicted in the two domains against its position with the unhindered cases again represented by the blue and magenta lines. Again, the velocity graphs for the same force overlap with each other due to the lack of any wall effects. It can be seen here that, in the unhindered case, the particle steadily approaches the wall with no change in velocity after it reaches its terminal velocity. After having attained the terminal velocity, the particle velocity oscillates about this terminal value because of the Brownian impulses experienced by the particle. The magnitude of the attracting force applied determines the magnitude of the terminal velocity attained, with the higher force producing a terminal velocity of higher magnitude.

On the other hand, as previously seen in the hindered cases, the approach of the particle toward the wall is influenced by the distance to the wall. It can be seen from both the position and velocity graphs that the initial approach, especially in the $10r_p$ and $30r_p$ domains, is similar to that in the unhindered case, since the influence of the wall is minimal in the center of these domains. However, as the wall gets closer, the hydrodynamic hindrance to the particle motion increases, causing the particle velocity to decrease continuously. The particle attains a new terminal velocity close to the wall once it crosses the final sampling point for the memory kernel close to the wall. As discussed previously, this depicts a lack of information close to the wall due to insufficient sampling of memory kernels close to the wall, which may be alleviated by extending the methodology. The hindrance is expected to keep increasing until the particle collides with the wall and, therefore, the particle velocity should ideally keep decreasing as the particle approaches. Nevertheless, at sufficiently small particle–wall separations, the continuum hypothesis breaks down and the hydrodynamic description is no longer valid. For the purpose of the present work, we refrain from further probing the exact limits of the hydrodynamic description of the closest approach to the wall. In the developed method, the increased hindrance effects due to the presence of the wall are effectively conveyed to the particle motion through the changes in the memory kernel as the particle position changes. The increase in the peaks of the memory kernel, as seen in Figs. 5 and 6, causes an increase in the hydrodynamic friction or hindrance experienced by the particle as the particle approaches the wall. Analytical force models, conventionally used in LPT, require elaborate expressions for how all the different components of the total hydrodynamic force (drag, added mass, history force, etc.) change with approach to the wall. Neglecting the hindrance effect altogether leads to significant errors in the predictions of the particle deposition timing, as well as consecutive errors in the predicted particle diffusion in the plane parallel to the wall, and thus to incorrect impact radii predictions.

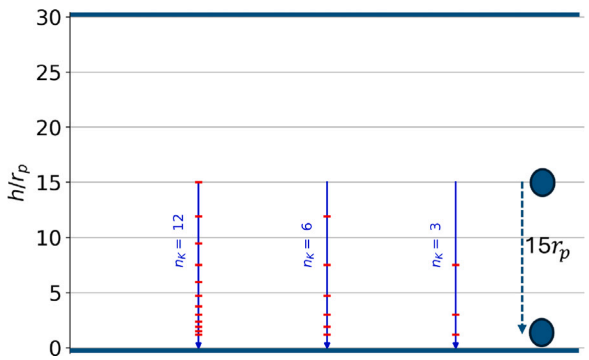


**Fig. 11.** Illustration of the memory kernel sampling locations. The blue lines indicate the main direction of motion of the Brownian particle during the deposition process. The particle starts at the center of the domain and then gradually approaches the wall under the influence of the steady force. The red marker along the path indicates the positions at which the memory kernel has been sampled to create the kernel library used to simulate the particle motion. The sampling position close to the wall has been used as the first sampling point in all the cases and the remaining sampling positions have been spaced out appropriately to get the required number of kernels. Here, $h$ is the distance from the bottom wall to the various positions in the domain.

In Fig. 10, the average normalized radii of impact in the hindered and unhindered cases are compared for the different domains. The unhindered cases are represented by the blue bar for the force of $m_pg$, while the red bar represents the same for the force of $5m_pg$. In the $5r_p$ and $10r_p$ domains, regardless of the attracting force, the impact radius in the unhindered case is smaller than that in the corresponding hindered case. In the $30r_p$ domain, when the attracting force is set to $m_pg$, the impact radius in the unhindered case is lower than that in the hindered case. When the attracting force is increased to $5m_pg$ in the largest domain, the impact radius is approximately the same, regardless of the hindrance experienced.

The apparent lack of a clear pattern in the dependence of the impact radius on the chosen combination of domain size and conservative force magnitude can be understood by looking at the factors influencing the impact radius. More specifically, the impact radius depends on the hindrance experienced by the particle, with a lower hindrance allowing the particle to diffuse further away from its initial position, thus effectively leading to a less focused impact region. On the other hand, the impact radius increases with the time it takes for the particle to reach the wall. The longer the migration time inside the microfluidic system, the longer the particle has to diffuse away from its deterministic (non-Brownian) position in the $xy$ plane. In the unhindered case, hindrance is low, and there is thus less resistance to parallel diffusion but also a faster approach to the wall, which reduces the time available for the particle to undergo wall-parallel diffusion. In contrast, in the hindered simulations, the particle experiences more resistance to motion, which reduces the diffusion of the particle while increasing the time needed to arrive at the wall, allowing the particle to spend more time diffusing. In conclusion, while one factor tends to increase the impact radius, the other tends to reduce it. The balance between these two opposing effects (the particle diffusion time in the system and the effective parallel diffusion coefficient) in each of the cases determines how close the impact radius in the unhindered case is to the same in the hindered case.

### *3.5. Trade-off between accuracy and computational cost in the resolution of the memory kernel*

The memory kernels used to create the Brownian motion simulations have been strategically determined as described in Section 2.3, based on the theoretical expectation of how the distance to the wall will influence the added mass effect. A total of 12 kernels are determined using IB-FSI simulations to create the memory kernel library for the nonlocal multiscale simulations of the separation of Brownian particles in a microfluidic device. In this section, the number of kernels ($n_K$) used to run the particle migration simulations is reduced from 12 to 6 and 3, respectively, to study the effect of the number of kernels on the particle motion. These simulations were run in the domain corresponding to a wall distance of $30r_p$, with the attracting force maintained at $m_pg$ to allow for the maximum amount of time for the particle to diffuse during motion in the $z$ direction. The memory kernels required for the simulations corresponding to $n_K = 6$ and $n_K = 3$ are taken as shown in Fig. 11. Here, the memory kernel corresponding to the position closest to the wall is kept constant while the remaining kernels are taken with the appropriate even spacing while moving towards the center to get the required number of kernels.

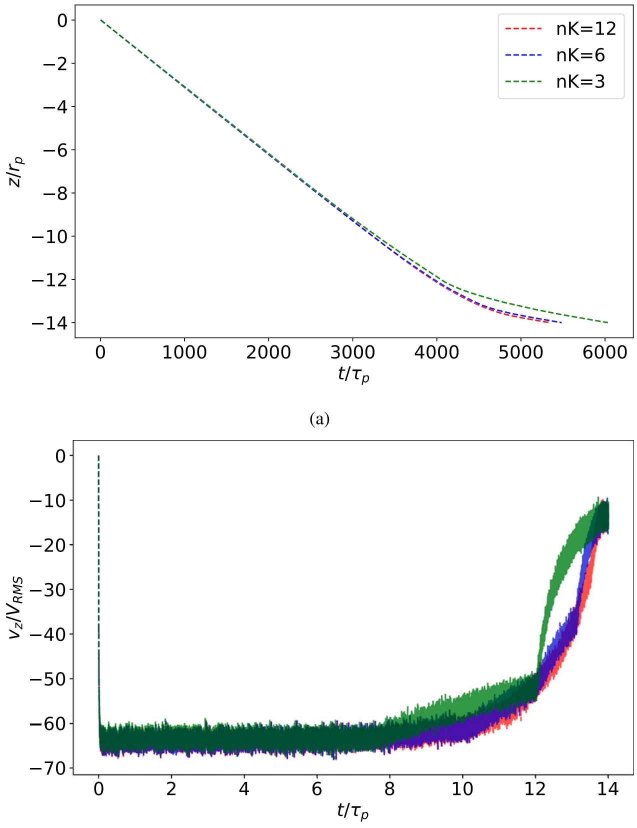
In Figs. 12(a) and 12(b), the changes in the position and velocity of the particle in the $z$ direction are depicted, as the number of kernels used to simulate the particle motion is varied. Here, the simulation consisting of 12 memory kernels is considered to be the most accurate, since it contains more sampling points to accurately represent the hydrodynamic effect. From Fig. 12(a), it can be seen that the particle position begins to deviate in the 6-kernel and 3-kernel version of the simulations from that seen in the 12-kernel version of the simulation in the near-wall region. The most drastic deviation is seen in the 3-kernel version of the simulation, since the wall-adjacent kernel begins to influence the behavior of the particle much earlier in the domain than in the other cases. Similarly, in the velocity plot shown in Fig. 12(b), the particle initially attains the same terminal velocity and then gradually shifts towards the final terminal velocity with which it strikes the wall (the same in all three cases). The shift in the particle velocities, however, depends on the number of kernels used. While the 12-kernel and 6-kernel versions show quite smooth transitions from the initial velocity to the final one, the 3-kernel version exhibits a sharp shift as it approaches close to the wall, indicating a sudden strong influence from the wall-adjacent kernel from that point forward. The 12-kernel version shows a slightly smoother approach to the final terminal velocity compared to the 6-kernel version, but the differences are minor compared to the drastic shift observed in the 3-kernel case.

Since the 12-kernel simulation run is considered the most accurate, the trajectory of the particle obtained from the 3-kernel and 6-kernel simulations are compared with the 12-kernel simulation to estimate the error that builds up in these simulations due to the lower number of kernels available. The error is calculated as shown in Eq. (21). Here, $x_{n_K,i}$ refers to the position of the particle in the simulation using $n_K$ kernels to simulate the particle motion. The numerator in this equation estimates the distance by which the particle deviates from the corresponding 12-kernel simulation, while the denominator is the total distance that the particle travels in the 12-kernel simulation.

$$Error(n_K) = \frac{\sum_{i=0}^{n} |(x_{n_K,i+1} - x_{n_K,i}) - (x_{12,i+1} - x_{12,i})|}{\sum_{i=0}^{n} |x_{12,i+1} - x_{12,i}|} \quad (21)$$

The error estimated in this manner for each direction is converted to a corresponding percentage reading as shown in Table 2. It can be seen that the simulation using 6 kernels shows a minimal error (less than 1%) in all three directions. On the other hand, for the simulation using 3 kernels, the error in the $z$ direction is more than 4%, indicating a larger deviation. Naturally, the error in a general scenario will depend on the system size, with more confined geometries leading to larger errors as the memory kernel resolution is decreased.

The simulation using 6 kernels shows less error than the same simulation using 3 kernels in all three directions. In general, the error is more pronounced in the wall-normal direction where the steady force is applied. While the steady push in this direction probably contributes to the increased error, it is also good to note that this is the direction in which the memory kernel shape and the associated hydrodynamic hindrance show the highest change.



(a)

(b)

**Fig. 12.** (a) The position of a particle in the $z$ direction as it undergoes Brownian motion while moving towards a wall under the influence of a steady force is plotted against time. (b) The velocity of the same particle in the $z$ direction is plotted against its position. A force equivalent to $m_pg$ is used to move the particle, while the number of memory kernels used to create the library is varied, $n_K \in [12, 6, 3]$.

**Table 2**
Error in position for different numbers of memory kernels used to create the simulations. The error has been estimated by comparing against the simulation that is run using 12 memory kernels.

| Number of Kernels | Error in position (%) | | |
|---|---|---|---|
| | x | y | z |
| 6 | 0.12 | 0.11 | 0.92 |
| 3 | 0.38 | 0.38 | 4.45 |

### 3.6. Proof of computational efficiency

The computational cost required to run the complete settling particle simulation involving Brownian motion is discussed in this section. The simulation in the domain with a wall distance of $30r_p$ with a force of $m_pg$ is studied using different computational setups to understand the cost in terms of core hours used to complete the simulations. Here, the multiscale simulation cost also includes the cost required to optimize and generate all the memory kernels required to run the simulations.

The costs for the different types of simulations are normalized using the total cost for the pure IB-FSI framework simulation. These are displayed in Fig. 13, where each bar represents the cost of a different simulation process. The yellow parts represent the cost of running the DNS using IB-FSI framework, either for the settling process or for determining the memory kernels. The blue parts represent the cost associated with running the LPT part of the multiscale model, using the memory kernels developed from the IB-FSI framework. It can be

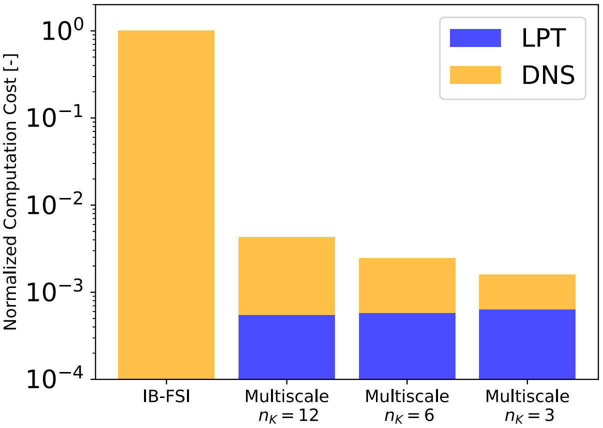


**Fig. 13.** Comparison between computational costs associated with running a pure DNS based on IB-FSI framework and the corresponding multiscale simulation using the memory kernel-based LPT framework. Here, $n_K$ refers to the number of memory kernels used to characterize the hydrodynamic effects the fluid in this particular domain has on the particle.

seen here that the cost of running the simulation purely using the IB-FSI framework is more than two orders of magnitude higher than that associated with the multiscale models, highlighting how much cheaper the currently proposed method is in comparison. Within the multiscale method, the computational cost required to prepare the kernels required for the simulations changes proportional to the number of kernels used to run the simulations, with the 12-kernel simulations requiring the largest preparatory cost and the 3-kernel simulations needing the smallest. The LPT components of the multiscale runs themselves do not show a significant difference between themselves, indicating that this cost depends more on the number of time steps required to reach the wall than on the number of kernels used.

The multiscale model using memory kernels thus provides a significant computational improvement over the pure IB-FSI simulations involving Brownian motion. Within the multiscale model, the number of times the IB-FSI simulations are used to determine the memory kernel becomes a differentiating factor. However, as discussed in the previous section, more kernels also provide more accurate hydrodynamic information about the flow field around the particle. In more generic flow scenarios, it might be useful to determine the memory kernels on-the-fly instead of pre-determining a kernel library for the whole particle flow path (as the flow path may be difficult to assess *a priori*). This scenario could be handled by interlinking the LPT simulation with the IB-FSI simulation, so that the multiscale model jumps from the LPT simulation to the IB-FSI simulation whenever it requires the determination of a new memory kernel. An adaptive determination of the kernel in this manner could be based on the rate of change of just the peak of the kernels as the particle moves through the domain. This type of kernel peak determination is relatively less computationally expensive as it involves just a single term compared to the determination of multiple values in the extended kernel. In such a scenario, it becomes important to strike a balance between the number of kernels utilized for the particle motion and the cost of determining all the kernels. An alternative approach would be to create a machine-learning model that can predict the memory kernel associated with a particle based on a limited dataset of sampled memory kernels. In such a case, the total cost of the multiscale model would also include the training time required to develop the machine-learning model. An exploratory study in this direction is presented in the next section.

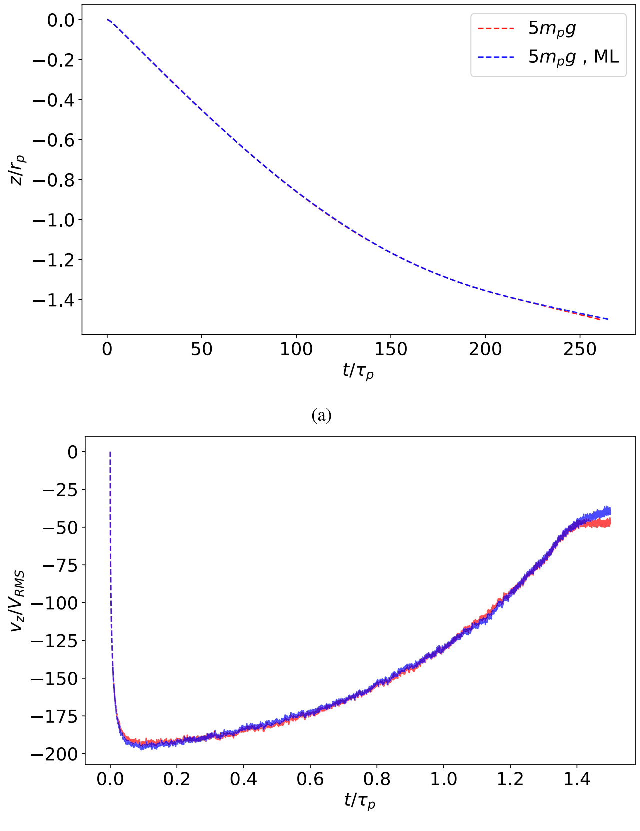
### 3.7. Towards a machine learning approach

The memory kernels obtained from the $5r_p$ domain are used to create a machine learning approach for predicting unknown memory kernels. Two Neural Network (NN) models are trained to predict the peaks and tails of the memory kernel in the chosen domain. The two models take the distance of the particle from the wall ($h$) as input and then produce the corresponding peaks or tails of the memory kernels, in both wall-normal and parallel directions. In the $5r_p$ domain, the memory kernel contains information up to a lag time of $10\tau_p$ and with a time step of $\Delta t = \tau_p/50$, producing time-discretized memory kernels consisting of $M = 500$ values. The peak of the kernel then refers to the first four values of the time-discretized memory kernel. While the tail refers to the remaining values. The models are created using the `PyTorch` and the `scikit-learn` modules in python (Paszke et al., 2017). The $5r_p$ domain is chosen since it has the highest variance in terms of kernel values compared to the other two domains, where the kernels are relatively constant in the center of the domain. This higher variety of kernels allows the models to learn more from the low number of kernels available.

Two NN models are chosen to predict the kernel, since a single model was observed to have difficulties in accurately predicting the whole kernel as the kernel values span multiple orders of magnitude between the peaks and the tails (as can be seen in Figs. 6 and 7). The Mean Square Error (MSE) is chosen as the loss function between the training data and the predicted data. The Adam optimization routine is used to tune the parameters of the models based on the loss function (Kingma and Ba, 2014). Both NN models consist of three layers of neurons connecting the input and output layers. The major difference between the models comes in the pre-processing employed on the training data. In the case of the model used to train the peaks, the training data is rescaled around the mean of all the values and normalized by their standard deviation. The training routine is then run for 1000 iterations until the parameters are tuned such that the MSE is less than $10^{-3}$, when rescaled back to the original data range. In the case of the NN model used for predicting the tails, the kernel tails used for training are converted to their logarithmic equivalents instead, to reduce the range over which the data spans. The model is then trained using these logarithmic values, and the final output is inversely transformed to obtain the actual predicted kernel tails. The training routine in this case is also run for 1000 iterations, until the MSE in the original data range is less than $10^{-4}$.

A simulation of a particle migrating in the $5r_p$ domain under the influence of a force equal to $5m_pg$ is run using the NN models to supply the required memory kernels. The two trained models generate the peaks and tails of memory kernels based on the distance of the particle from the bottom wall. Combining the peak and tail gives the complete kernel required to run simulations in the $5r_p$ domain. The position and velocity graphs corresponding to this simulation are compared with those obtained from the simulations run using the linear interpolation between known kernels in Fig. 14. Both the position and velocity graphs from utilizing the machine learning approach can be seen to closely follow the graphs of the same data from the simulations based on linear interpolation . An estimation of the error in the trajectory was done using Eq. (21). Along the $x$ and $y$ directions, the machine learning approach shows a deviation of less than 0.5% while in the $z$ direction, it shows a deviation of approximately 1.3%. These small deviations indicate that memory kernels predicted using machine learning methods can be used to produce accurate hydrodynamic simulations. The machine learning model is also able to predict the kernels that produce higher hydrodynamic hindrance, as seen from the decrease in the particle velocity beyond the final kernel sampling point. Thus, this approach allows for some extrapolation that was previously avoided in the linear interpolation approach towards kernel prediction. From running a pure settling simulation and comparing to its DNS equivalent as done in the validation cases, the deviation beyond the final kernel sampling point in this case is found to reduce to 1.65% compared to the 2.3% in the linear interpolation case where the kernel is held constant.

While this investigation confirms the possibility for such an approach, various factors such as the pre-processing method, the neural



(a)

(b)

**Fig. 14.** The position and velocity graphs for the particle migration simulations utilizing the NN models are depicted here. The simulations take place in the $5r_p$ domain under the influence of a steady force of $5m_pg$. Figure (a) shows the position plotted against the time while figure (b) shows the velocity plotted against the position, all along the $z$ direction.

architecture, optimization routine, loss function etc., used while training the models can be altered to get better predictions of the kernels. In the case of more complex particle and domain geometries (such as non-spherical particles or branched configurations in microfluidic devices), the input parameters can be increased from just the particle position to include factors such as the particle size, orientation, domain size and so on. Multiparticle systems could include relative positions and orientations between particles as possible input parameters. If the fluid used in the simulations is altered, then parameters such as the fluid–particle density ratio and fluid viscosity can also be included as input parameters for the models. The output parameters in this case are the wall-normal and parallel kernels but these can be altered to include three different directions to incorporate varying hydrodynamic effects in each direction. The model could also be extended to include off-diagonal components of the memory kernel in asymmetric particle or domain shapes, where such terms can become significant. Rotational Brownian motion might also be relevant in the case of such non-spherical particles, in which case, similar memory kernels will need to be derived for the rotational motion of the particle as well. Here, the angular velocity history of the particle and hydrodynamic torque experienced by the particle can be used to optimize rotational memory kernels in a manner similar to that detailed in this work for the translational motion. These models will, of course, require initial training data that spans a wide range of these input and output parameters so that the models can be as generic as possible. A possible future avenue for this modeling approach thus involves the generation of this data and training of such models for either case-specific needs or general application.

## 4. Conclusions

In this work, a novel nonlocal multiscale model has been developed and applied for simulations of Brownian microfluidic systems. The model is based on the generation of hydrodynamic memory kernels using a multiphase Direct Numerical Simulation (DNS) framework, which are subsequently used in a Lagrangian Particle Tracking (LPT) routine to evolve particle trajectories. The model is based on the Generalized Langevin Equation (GLE) and generates the complete hydrodynamic force, as well as the corresponding random colored Brownian force, using the hydrodynamically optimized memory kernels. In this way, the model does not require any analytical models for the various force contributions or their dependence on the vicinity of nearby walls or surfaces, thereby allowing for its utilization in complicated geometric configurations where such analytical models for the hydrodynamic model are hard to derive. The method can simultaneously generate random colored forces to incorporate Brownian motion into the particle dynamics, thus making it especially suited to study particulate flows at the micro- and nano-scales. The model is validated for canonical cases involving acceleration from a steady force and unhindered Brownian motion. Particle settling simulations are also used to validate that linear interpolation between known values of the memory kernels is sufficient to capture the hydrodynamic behavior in intermediate positions.

The validated model is thereafter used to study particle migration and deposition in microfluidic systems, where a conservative force attracts particles to a target wall while the particle undergoes Brownian motion. The impact radius is determined as a measure to quantify the capability of a given system design (combination of geometrical factors and magnitude of the attracting force) to focus particles into a narrow region. It is shown that both hindrance effects and nonlocal effects are significant in microfluidic systems for neutrally buoyant particles. Moreover, it is shown how balancing the choice of system size and the magnitude of the attracting force can be leveraged to optimally focus particle deposition locations on the target wall. These effects arise as the hindering force experienced by the particles influences the approach of the particle towards the wall (and the duration of the approach), as well as the diffusion the particle undergoes in the direction perpendicular to the wall. The dependence of the simulations on the number of known kernels is demonstrated by varying the number of kernels used to simulate the particle migration, with more known kernels providing more accurate results. Finally, a machine learning model trained on the known kernels is shown to be capable of predicting accurate memory kernels necessary for running similar particle migration simulations.

The memory-kernel-based LPT method proposed here is thus demonstrated to be a suitable low-cost tool to assist in the study of flow phenomena involving Brownian motion in confined systems. The method can be further extended to incorporate various particle shapes and rotation, since the memory kernel inherently carries such information, and is also well suited for combination with machine-learning approaches to memory-kernel prediction on the basis of limited DNS data.

## CRediT authorship contribution statement

**Anand Joseph Michael:** Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Formal analysis, Conceptualization. **Andreas Mark:** Writing – review & editing, Supervision, Software, Methodology, Conceptualization. **Srdjan Sasic:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Henrik Ström:** Writing – review & editing, Supervision, Software, Resources, Project administration, Methodology, Funding acquisition, Conceptualization.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Henrik Strom reports financial support was provided by Swedish Research Council. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This work has been financed by the Swedish Research Council (Vetenskapsrådet) via grant agreement Dnr 2021-05175. The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council (Vetenskapsrådet) through grant agreement no. 2022-06725.

## Data availability

Data will be made available on request.

## References

Arefi, S., 2021. Computational Studies of Particles and Cells Transport in Microfluidic Devices (Ph.D. thesis). University of British Columbia.

Arefi, S.M.A., Yang, C.W.T., Sin, D.D., Feng, J.J., 2020. Simulation of nanoparticle transport and adsorption in a microfluidic lung-on-a-chip device. Biomicrofluidics 14, 044117.

Bian, X., Kim, C., Karniadakis, G.E., 2016. 111 years of Brownian motion. Soft Matter 12 (30), 6331–6346.

Bonilla, F.A., Kleinfelter, N., Cushman, J.H., 2007. Microfluidic aspects of adhesive microbial dynamics: A numerical exploration of flow-cell geometry, Brownian dynamics, and sticky boundaries. Adv. Water Resour. 30, 1680–1695.

Byrd, R., Lu, P., Nocedal, J., Zhu, C., 1995. A limited memory algorithm for bound constrained optimization. SIAM J. Sci. Comput. 16 (5), 1190–1208.

Cejas, C.M., Maini, L., Monti, F., Tabeling, P., 2019. Deposition kinetics of bi- and tridisperse colloidal suspensions in microchannels under the van der waals regime. Soft Matter 15, 7438–7447.

Cejas, C.M., Monti, F., Truchet, M., Burnouf, J.-P., Tabeling, P., 2017. Particle deposition kinetics of colloidal suspensions in microchannels at high ionic strength. Langmuir 33, 6471–6480.

Çetin, B., Li, D., 2011. Dielectrophoresis in microfluidics technology. Electrophoresis 32 (18), 2410–2427.

Choi, S., Lee, W.I., Lee, G.H., Yoo, Y.-E., 2020. Analysis of the binding of analyte-receptor in a micro-fluidic channel for a biosensor based on Brownian motion. Micromachines 11, 570.

Daitche, A., 2015. On the role of the history force for inertial particles in turbulence. J. Fluid Mech. 782, 567–593.

De Santo, I., D'Avino, G., Romeo, G., Greco, F., Netti, P.A., Maffettone, P.L., 2014. Microfluidic Lagrangian trap for Brownian particles: Three-dimensional focusing down to the nanoscale. Phys. Rev. Appl. 2, 064001.

deMello, A.J., 2006. Control and detection of chemical reactions in microfluidic systems. Nature 442, 394–402.

Díaz, M.V., 2021. On the long-time persistence of hydrodynamic memory. Eur. Phys. J. E 44 (11), 141.

Fan, L., Wu, Y., Jian, E., Tseng, J., Wan, R., Tseng, A., Lin, J., Lee, E., 2022. Diffusiophoresis of a highly charged dielectric fluid droplet induced by diffusion potential. Phys. Fluids 34 (4).

Felderhof, B., 2005. Effect of the wall on the velocity autocorrelation function and long-time tail of Brownian motion. J. Phys. Chem. B 109 (45), 21406–21412.

Forbes, T.P., Forry, S.P., 2012. Microfluidic magnetophoretic separations of immunomagnetically labeled rare mammalian cells. Lab A Chip 12 (8), 1471–1479.

Furlani, E.P., 2010. Magnetic biotransport: analysis and applications. Materials 3 (4), 2412–2446.

Haghparas, Z., Tabalvandani, M.B., Arghavani, P., Hosseini, S.B., Badieirostami, M., Habibi-Rezaei, M., Moosavi-Movahedi, A.A., 2025. Modeling the navigating forces behind BSA aggregation in a microfluidic chip. Soft Matter 21, 989–1001.

Hale, M.S., Mitchell, J.G., 2001. Motion of submicrometer particles dominated by Brownian motion near cell and microfabricated surfaces. Nano Lett. 1 (11), 617–623.

Hauge, E., Martin-Löf, A., 1973. Fluctuating hydrodynamics and Brownian motion. J. Stat. Phys. 7, 259–281.

van Hinsberg, M.A.T., ten Thije Boonkamp, J.H.M., Clercx, H.J.H., 2011. An efficient, second order method for the approximation of the Basset history force. J. Comput. Phys. 230, 1465–1478.

Jaganathan, D., Prasath, S.G., Govindarajan, R., Vasan, V., 2023. The Basset–Boussinesq history force: its neglect, validity, and recent numerical developments. Front. Phys. 11, 1167338.

Jung, G., Hanke, M., Schmid, F., 2017. Iterative reconstruction of memory kernels. J. Chem. Theory Comput. 13 (6), 2481–2488.

Jung, G., Hanke, M., Schmid, F., 2018. Generalized langevin dynamics: construction and numerical integration of non-Markovian particle-based models. Soft Matter 14 (46), 9368–9382.

Kannan, A.S., Mark, A., Maggiolo, D., Sardina, G., Sasic, S., Ström, H., 2021. Assessment of hindered diffusion in arbitrary geometries using a multiphase DNS framework. Chem. Eng. Sci. 230, 116074.

Kannan, A.S., Naserentin, V., Mark, A., Maggiolo, D., Sardina, G., Sasic, S., Ström, H., 2019. A continuum-based multiphase DNS method for studying the Brownian dynamics of soot particles in a rarefied gas. Chem. Eng. Sci. 210, 115229.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kroslak, M., Sefcik, J., Morbidelli, M., 2007. Effects of temperature, pH, and salt concentration on $\beta$ -Lactoglobulin deposition kinetics studied by optical waveguide lightmode spectroscopy. Biomacromolecules 8, 963–970.

Lauga, E., Squires, T., 2005. Brownian motion near a partial-slip boundary: A local probe of the no-slip condition. Phys. Fluids 17, 103012.

Lee, Y.K., Porter, C., Diamond, S.L., Crocker, J.C., Sinno, T., 2018. Deposition of sticky spheres in channel flow: Modeling of surface coverage evolution requires accurate sphere-sphere collision hydrodynamics. J. Colloid Interface Sci. 530, 383–393.

Li, A., Ahmadi, G., 1992. Dispersion and deposition of spherical particles from point sources in a turbulent channel flow. Aerosol Sci. Technol. 16 (4), 209–226.

Lin, Y.-J., Cao, T., Chacón-Patiño, M.L., Rowland, S.M., Rodgers, R.P., Yen, A., Biswal, S.L., 2019. Microfluidic study of the deposition dynamics of asphaltene subfractions enriched with island and archipelago motifs. Energy & Fuels 33, 1882–1891.

Liu, D., Nocedal, J., 1989. On the limited memory method for large scale optimization. Math. Program. B 45 (3), 503–528.

Liu, H., Pahlavan, A.A., 2025. Diffusioosmotic reversal of colloidal focusing direction in a microfluidic T-junction. Phys. Rev. Lett. 134, 098201.

Lyklema, J., van Leeuwen, H.P., Minor, M., 1999. DLVO-theory, a dynamic re-interpretation. Adv. Colloid Interface Sci. 83, 33–69.

Ma, H., Pazmino, E.F., Johnson, W.P., 2011. Gravitational settling effects on unit cell predictions of colloidal retention in porous media in the absence of energy barriers. Environ. Sci. Technol. 45 (19), 8306–8312.

Mainardi, F., Mura, A., Tampieri, F., 2010. Brownian motion and anomalous diffusion revisited via a fractional langevin equation. arXiv preprint arXiv:1004.3505.

Mark, A., Rundqvist, R., Edelvik, F., 2011. Comparison between different immersed boundary conditions for simulation of complex fluid flows. Fluid Dyn. Mater. Process. 7 (3), 241–258.

Mark, A., van Wachem, B.G., 2008. Derivation and validation of a novel implicit second-order accurate immersed boundary method. J. Comput. Phys. 227 (13), 6660–6680.

Michael, A.J., Mark, A., Sasic, S., Ström, H., 2025. Generalized langevin dynamics in multiphase direct numerical simulations using hydrodynamically optimized memory kernels. Phys. Fluids 37 (3).

Michaelides, E.E., 2016. Wall effects on the Brownian movement, thermophoresis, and deposition of nanoparticles in liquids. J. Fluids Eng. 138, 051303.

Midelet, C., Le Piofle, B., Werts, M.H.V., 2019. Brownian motion and large electric polarizabilities facilitate dielectrophoretic capture of sub-200 nm gold nanoparticles in water. ChemPhysChem 20, 3354–3365.

Mohammadghasemi, H., Mozaffari, S., Kalfati, M.S., Ghasemi, H., Nazemifard, N., 2024. Prediction of asphaltene deposition dynamics in various microfluidic geometries using computational fluid dynamics. Energy & Fuels 38, 7786–7800.

Mustin, B., Stoeber, B., 2010. Deposition of particles from polydisperse suspensions in microfluidic systems. Microfluid. Nanofluidics 9, 905–913.

Newmark, N.M., 1962. A method of computation for structural dynamics. Trans. Am. Soc. Civ. Eng. 127 (1), 1406–1433.

Ounis, H., Ahmadi, G., McLaughlin, J.B., 1991. Brownian diffusion of submicrometer particles in the viscous sublayer. J. Colloid Interface Sci. 143 (1), 266–277.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A., 2017. Automatic differentiation in pytorch.

Python Software Foundation, 2025. The python standard library: random — Generate pseudo-random numbers. URL https://docs.python.org/3/library/random.html.

Qiu, Y., Lu, C., Bao, F., Hu, G., 2023. Design of a multilayer lung chip with multigenerational alveolar ducts to investigate the inhaled particle deposition. Lab A Chip 23, 4302–4312.

Ruggeri, F.S., Charmet, J., Kartanas, T., Peter, Q., Chia, S., Habchi, J., Dobson, C.M., Vendruscolo, M., Knowles, T.P.J., 2018. Microfluidic deposition for resolving single-molecule protein architecture and heterogeneity. Nat. Commun. 9, 3890.

Shin, S., 2020. Diffusiophoretic separation of colloids in microfluidic flows. Phys. Fluids 32 (10).

Simha, A., Mo, J., Morrison, P., 2018. Unsteady Stokes flow near boundaries: the point-particle approximation and the method of reflections. J. Fluid Mech. 841, 883–924.

Srinivasan, S., Van den Akker, H.E.A., Shardt, O., 2022. Inclusion of DLVO forces in simulations of non-Brownian solid suspensions: Rheology and structure. Int. J. Multiph. Flow 149, 103929.

Sznitman, J., 2022. Revisiting airflow and aerosol transport phenomena in the deep lungs with microfluidics. Chem. Rev. 122, 7182–7204.

Unni, H.N., Yang, C., 2005. Brownian dynamics simulation and experimental study of colloidal particle deposition in a microchannel flow. J. Colloid Interface Sci. 291, 28–36.

Unni, H.N., Yang, C., 2006. Deposition of colloidal particles from pressure driven microfluidic flow – Brownian dynamics simulations. Comput. Methods 2, 1441–1445.

Unni, H.N., Yang, C., 2009. Colloidal particle deposition from electrokinetic flow in a microfluidic channel. Electrophoresis 30, 732–741.

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors, 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods 17, 261–272. http://dx.doi.org/10.1038/s41592-019-0686-2.

Yan, Z., Huang, X., Yang, C., 2025. Deposition of colloidal particles in a microchannel at elevated temperatures. Microfluid. Nanofluidics 18, 403–414.

Zhu, C., Byrd, R., Lu, P., Nocedal, J., 1997. L-BFGS-B: Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization. ACM Trans. Math. Software 23 (4), 550–560.