

Received 15 July 2024, accepted 31 July 2024, date of publication 5 August 2024, date of current version 16 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3438243

METHODS

OpenFOAM Simulation of Microfluidic Molecular Communications: Method and Experimental Validation

PIT HOFMANN¹, (Graduate Student Member, IEEE), PENGJIE ZHOU¹,
CHANGMIN LEE^{2,3}, (Member, IEEE), MARTIN REISSLEIN⁴, (Fellow, IEEE),
FRANK H. P. FITZEK^{1,5}, (Senior Member, IEEE), AND
CHAN-BYOUNG CHAE², (Fellow, IEEE)

¹Deutsche Telekom Chair of Communication Networks, Technische Universität Dresden, 01062 Dresden, Germany

²School of Integrated Technology, Yonsei University, Seoul 03722, South Korea

³Newratek Inc, Seoul 06715, South Korea

⁴School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287, USA

⁵Centre for Tactile Internet with Human-in-the-Loop (CeTI), 01069 Dresden, Germany

Corresponding authors: Martin Reisslein (reisslein@asu.edu) and Chan-Byoung Chae (cbchae@yonsei.ac.kr)

This work was supported in part by the German Research Foundation (DFG) through Germany's Excellence Strategy–EXC 2050/1–Cluster of Excellence “Centre for Tactile Internet with Human-in-the-Loop” (CeTI) of Technische Universität Dresden under Project ID 390696704; in part by the Federal Ministry of Education and Research (BMBF) through the Program of “Souverän. Digital. Vernetzt.” Joint Project 6G-life under Grant 16KISK001K; in part by the Project Internet of Bio-Nano Things (IoBNT) under Grant 16KIS1994; and in part by the National Research Foundation of Korea (NRF) through the Ministry of Science and ICT, Korean Government under Grant RS-2023-00208922, Grant RS-2024-00428780, and Grant 2022R1A5A1027646.

ABSTRACT The convergence of microfluidics and Molecular Communication (MC) has garnered significant attention in the context of the emerging Internet of Bio-Nano Things (IoBNT) paradigm. Microfluidic systems offer unique capabilities for manipulating fluids at the microscale, enabling precise control over molecular interactions. Potential IoBNT applications of microfluidic MC lie in healthcare, environmental monitoring, and biotechnology. The MultiPhase Particle-In-Cell (MPPIC) solver in OpenFOAM was originally intended for applications in sedimentation, separation, and granular flows. In contrast, in this methods article, we investigate the feasibility of utilizing the MPPIC solver in OpenFOAM for the simulation of microfluidic MC in the context of the IoBNT. This methods article thus provides a simulation framework for studying the intricate interplay between microfluidics and MC; the examined OpenFOAM simulation methodology can serve as a foundation for further research in the field of microfluidic MC simulation. Specifically, we compare our OpenFOAM simulations to an analytical model as well as to a particle-based simulation approach. For a uniform release of particles on the transmitter side and a flow-dominated regime, our simulation results demonstrate that the Channel Impulse Response (CIR) in the analytical approach and in the particle-based approach are similar to the OpenFOAM simulation results. As the geometry expands in the direction of the fluid flow, the OpenFOAM simulations converge to the analytical model, especially for the tail of the CIR. In addition, we compare the OpenFOAM simulation results with measurements on a fluidic MC testbed in the millimeter scale. We find that the OpenFOAM simulation results closely approximate the experimental measurements for the flow-dominated regime with a Root-Mean-Square Error (RMSE) below 10 %.

INDEX TERMS Channel modeling, Internet of Bio-Nano Things, microfluidic systems, molecular communications, OpenFOAM, simulation, testbed verification.

The associate editor coordinating the review of this manuscript and approving it for publication was Tariq Umer¹.

I. INTRODUCTION

The rapid advancement of nanotechnology and the convergence of biology with nanoscale devices have given rise to the concept of the Internet of Bio-Nano Things (IoBNT) [1], [2], [3]. The IoBNT represents a paradigm shift in the field of communication and sensing, enabling seamless integration of biological entities and nanoscale devices into a networked ecosystem [4], [5], [6]. This transformative concept has opened new frontiers in healthcare, environmental monitoring, and biotechnology by allowing real-time monitoring, data exchange, and precise control of biological processes at the molecular level [1], [7].

A. MOTIVATION FOR SIMULATION OF MICROFLUIDIC MC

One of the pivotal branches of the IoBNT is microfluidic Molecular Communication (MC). Microfluidic MC involves the information and signal exchange through the transmission of molecules or particles within microfluidic channels [8]. Unlike conventional electronic communication systems, microfluidic MC exploits the principles of diffusion and advection to transfer information, offering unique advantages in biocompatibility, scalability, and energy efficiency [8], [9], [10]. Microfluidic MC has enabled numerous applications in biomedical engineering and targeted drug delivery [11], lab-on-a-chip technologies [12], [13], and synthetic biology [14], [15], revolutionizing the manipulation of biological systems with unprecedented precision.

To harness the full potential of microfluidic MC and the IoBNT, researchers have increasingly turned to simulation and computational modeling as indispensable tools [16]. The complexity of molecular interactions and the dynamic nature of microfluidic environments require advanced techniques to analyze, optimize, and design communication systems effectively. Simulation of microfluidic MC offers efficient and cost-effective means to study the behavior of particles within microfluidic networks and to explore various signaling schemes [16].

B. ORIGINAL CONTRIBUTIONS AND STRUCTURE

We investigate the use of OpenFOAM for the simulation of microfluidic MC. As reviewed in Section II-B2, OpenFOAM is a widely-used open-source tool for solving complex fluid flow problems through Computational Fluid Dynamics (CFD) simulations. More specifically, in this methods article, we investigate the MultiPhase Particle-In-Cell (MPPIC) solver, which is well-known in the OpenFOAM community as a two-phase solver. Our main contribution is to investigate OpenFOAM with the MPPIC solver as a new tool for the simulation of microfluidic MC. To the best of our knowledge, OpenFOAM with the MPPIC solver has not previously been studied for CFD simulation of microfluidic MC.

The MPPIC method accurately maps (bi-directionally) between Lagrangian particles and a computational grid [17]. Treating the phase of the particles as a fluid, derived continuum terms are evaluated on a grid and “mapped back to the

individual particles” [17]. By this method, a computational technique for multiphase flows can be implemented, which accommodates dense and diluted particle loading, different particle materials, or different particle size distributions [17]. Andrews and O’Rourke [17] envisioned applications of the MPPIC solver in sedimentation, separation, fluidized beds (e.g., combustion and catalytic cracking), and other granular flows. In contrast to these macro-scale applications that consider the handling of large quantities of materials in industrial environments, this methods article conducts a feasibility and validation study of using OpenFOAM with the MPPIC solver for simulating microfluidic MC.

We provide a novel validated framework for accurate simulations of microfluidic MC. Toward this goal, we compare the simulation results obtained with OpenFOAM utilizing the two-phase MPPIC solver for a flow-dominated regime with an analytical channel model and with Particle-Based Simulations (PBSs). Specifically, we evaluate the error between the simulation results obtained with OpenFOAM (using MPPIC) and the analytical model as well as the PBS. In addition, we utilize the experimental testbed setup proposed by Angerbauer et al. [18] for conducting experimental measurements, which we compare to the OpenFOAM simulation results that we obtained using the MPPIC solver in the millimeter scale.

To facilitate the use of OpenFOAM with the MPPIC two-phase solver for microfluidic MC simulations, we provide with this methods article a publicly accessible repository [19]. This repository provides a template (folder structure of scripts) for conducting microfluidic MC simulations using the OpenFOAM MPPIC. In addition, the repository provides Python code for the post-processing of OpenFOAM simulation results.

The rest of this methods article is structured as follows. Section II reviews the related literature on analytical modeling, simulation, and experimental testbeds for microfluidic MC. Section III explains the IoBNT scenario, models the system, and describes the fundamentals for microfluidic MC. The analytical approach for microfluidic MC, which we use as an evaluation benchmark, is reviewed in Section IV. Section V explains the simulation basics of microfluidic MC, especially regarding the MPPIC solver properties for the fluid phase, the solid phase (i.e., the particles), as well as the coupling between these phases. Section VI introduces the methodology for microfluidic MC, including pre-processing, hard- and software properties for the final computation, and the simulation parameters, such as the geometry and meshing, the properties of the fluid, the walls, and the particles, as well as the boundary conditions for velocity and pressure. Section VII presents the post-processing of the simulation results with regard to the Channel Impulse Response (CIR) and the associated error calculation. We compare the OpenFOAM MPPIC simulation results to an analytical approach, as well as to PBS results. Section VIII describes the utilized experimental testbed and compares the measurements from the testbed with the

OpenFOAM simulations. The results and limitations are discussed in Section IX. Finally, in Section X, a conclusion and outlook for future research directions follow.

II. RELATED WORK

The section reviews the related literature on MC evaluation methods, which fall into the main categories of analytical modeling, simulation, and measurements on experimental testbeds. The main purpose of this review is to distinguish our simulation method from the existing simulation methods. We also briefly review the existing analytical modeling and experimental testbed literature as we benchmark our simulation method against an analytical model and an experimental testbed.

A. ANALYTICAL MODELING

The field of analytically modeling microfluidic MC channels has received relatively little research attention to date. Bicen and Akyildiz [8] examined flow models within microfluidic channels of varying cross-sectional areas. These models involve the derivation of the CIRs through the solution of a One-Dimensional (1-D) diffusion-advection equation, which is only valid under specific conditions [8]. In [20], Bicen and Akyildiz analyzed the propagation noise and memory in flow-induced MC. The analysis examines the “end-to-end propagation noise” [20] influenced by the parameters of microfluidic channels. In 2015, Bicen and Akyildiz [21] analyzed the capacity and modeled the interference for microfluidic MC channels.

Wicke et al. [16] examined the CIRs of microfluidic channel systems in the context of both dispersion and flow-dominated regimes. In the dispersion scenario, an assumption is made regarding the uniform distribution of particles radially, leading to a reduction of the system to a 1-D differential equation for obtaining the CIR. Conversely, in the flow-dominated scenario, only the flow aspect is taken into consideration, while the diffusion effects are neglected. Dinc et al. [22] analytically derived the CIR of a microfluidic channel incorporating the *Poiseuille* flow for a point transmitter that is positioned arbitrarily within the channel, and an observing receiver whose planar geometry fully covers the channel’s cross-section. Jamali et al. [23] provided a tutorial on channel modeling for diffusive MC, including flow-induced MC, while Hamidović et al. [24] reviewed microfluidic systems for MC.

We utilize the analytical model of Wicke et al. [16] as a benchmark for evaluating our original simulation approach of using OpenFOAM with the MPPIC two-phase solver for microfluidic MC simulations.

B. SIMULATION

MC simulators fall into the two main categories of MC supporting simulators as well as MC & microfluidics supporting simulators [24]. While MC supporting simulators focus purely on Monte Carlo simulations, MC & microfluidics supporting simulations also consider the characteristics of the

fluid flow or fluid dynamics. We briefly review these existing MC simulators and contrast them from our MC simulation method.

1) MC SUPPORTING SIMULATORS

Jian et al. presented the MC supporting simulator nanoNS3 [25] which models bacterial MC networks with partial differential equations, such as Fick’s law, in the spatiotemporal domain [24]. The MC supporting simulators [26], [27], [28], [29] count discrete molecules in subvolumes that are structured according to a grid [24]. Specifically, Drawert et al. [26] utilized Unstructured triangular and tetrahedral meshes in conjunction with a Reaction-Diffusion Master Equation formalism to develop the URDME simulation method. Smartcell [27] simulates cellular processes in simple networks combining stochastic approximation with localization and diffusion [27]. Hattne et al. [28] presented the MesoRD simulator for the stochastic simulation of chemical reactions and diffusion. MesoRD implements the “next subvolume method” to simulate exactly the reaction-diffusion equation [28]. Wei et al. [29] presented the BNSim for the simulation of bacteria-based networks. BNSim integrates algorithms for simulation, genetic circuits, and pathways for bacterial chemotaxis in a Three-Dimensional (3-D) environment [29]. Turgut et al. [30] presented the Neural NaNoNetwork Simulator (N⁴Sim) as an event-based simulator. The N⁴Sim is designed specifically for the simulation of nervous systems, taking into account synaptic molecular communication.

The simulators in [31], [32], [33], [34], [35], [36], [37], and [38] track individual particles in space and time with a Particle-Based Simulation (PBS) approach [24]. Andrews et al. [31] developed a simulation tool in the Smoldyn program for cellular processes, including algorithms for modeling the effects of diffusion, interactions of the membranes, and individual molecule reactions. Felicetti et al. [32] developed the BiNS2 simulator for diffusion-based MC considering advection, i.e., additional drift, inside blood vessels. Llatser et al. [33], [34] presented N3Sim, a simulator also for diffusion-based MC. N3Sim allows the simulation of On-Off Keying (OOK) as a modulation technique. The movement of the released particles is modeled by Brownian motion, considering the particles’ inertia and interactions [33], [34]. Noel et al. [35] developed the Actor-based Communication via Reaction-Diffusion (AcCoRD) simulator, which utilizes reaction-diffusion equations in conjunction with hybrid models for microscopic and mesoscopic simulation [35]. Wang et al. [37] proposed an *a priori* algorithm for the simulation of absorbing receivers in diffusion-based MC, i.e., the fraction of particles absorbed by an absorbing, spherical receiver is simulated by using an *a priori* Monte Carlo approach. BioNetGen [36] is a software tool designed for rule-based modeling of complex biochemical systems. Finally, the MolecUlar CommunicatIoN (MUCIN) simulator [38] is an end-to-end simulation tool for diffusion-based

MC with additional drift, supporting 3-D environments. We use the MUCIN simulator [38] as a representative PBS simulator for benchmarking the OpenFOAM MPPIC solver.

2) MC & MICROFLUIDICS SUPPORTING SIMULATORS

MC & microfluidics supporting simulators encompass mainly the CFD simulators. Besides the OpenFOAM [39] software tool for CFD, COMSOL Multiphysics [40] and ANSYS Fluent [41] are two commonly used commercially available tools. For example, COMSOL Multiphysics was utilized in [42] and [43], while ANSYS Fluent was used in [44] and [45]. In addition, Fink et al. [46] proposed an efficient, freely-available software tool for the simulation of droplet-based microfluidic devices. Also, Matlab supports MC & microfluidics simulations, e.g., with the Partial Differential Equation (PDE) Toolbox, the CFD toolbox, and the Particle Tracking Toolbox [24]. A quantitative comparison of the simulation platforms is beyond [24] and beyond the scope of this paper and is an important direction for future research. In the following, we will focus on OpenFOAM, which we use as a software tool for the CFD MC simulation in this methods article.

Originally, OpenFOAM for MC was introduced by Drees et al. [47] in 2020 by proposing the *Pogona* simulator, simulating macroscale MC, e.g., in pipe networks. The fundamental assumption in [47] is perfect advection, wherein the particles' behavior mimics that of the surrounding liquid due to their small size, i.e., the solver in [47], named *icoFoam*, is a *one-phase* solver. Using the Pressure-Implicit with Splitting of Operators (PISO) algorithm [48], *icoFoam* solves the incompressible *Navier-Stokes* equations for the laminar regime. In [47], particle movement is determined by leveraging the fluid simulation's predicted liquid movement, which is pre-computed using a CFD simulator and exported as a vector field [47]. During each simulation time step, the particles' positions are updated by applying the flow information to move the particles toward their new positions, facilitating the propagation of information through a network of tubes by expelling particles at the transmitter and detecting their presence at the receiver [47].

In contrast to the one-phase solver in [47], we examine the use of MPPIC, which is a *two-phase* solver for microfluidic MC simulation. A *one-phase* solver, such as *icoFoam*, is designed to handle the dynamics of a single phase, e.g., a fluid phase, focusing on solving the incompressible *Navier-Stokes* equations. *One-phase* solvers are suitable for fluid flow simulations where interactions with other phases are absent. In contrast, a *two-phase* solver, such as MPPICFoam, can simulate the interactions between two distinct phases. A two-phase solver can accurately account for interaction phenomena, e.g., fluid-solid interactions [49], making it a powerful tool in CFD simulations. Therefore, a *two-phase* solver is more appropriate for microfluidic MC simulations (than a one-phase solver), especially when the interactions between the fluid and solid phases become relevant.

As groundwork towards enabling simulation of microfluidic MC simulation with phase interactions, we introduce and evaluate OpenFOAM with the two-phase MPPIC solver for microfluidic MC simulation in this methods article. We do not examine specific phase interactions in this article; rather, we enable the simulation evaluation of microfluidic MC with such phase interactions.

C. MEASUREMENTS IN EXPERIMENTAL TESTBEDS

We quantitatively compare our OpenFOAM simulation results with experimental testbed measurements in Section VIII. This subsection briefly reviews existing microfluidic MC testbeds to provide the context for the testbed that we use in our evaluations.

Kuscu et al. [50] fabricated a receiver based on graphene field-effect transistor biosensors for information exchange with encoding into concentration levels of single-stranded DNA molecules in a microfluidic MC testbed. Cali et al. [51], [52] employed fluorescent carbon nanoparticles as information carriers in a microfluidic MC experimental testbed where the information carriers are fluorescent carbon nanoparticles and the receiver is a fluorescence detector. Walter et al. [53] designed a liquid-based microfluidic MC experimental testbed where information is encoded into a concentration level of sodium hydroxide. Chemical reactions are used on the receiver side for the detection, thresholding, and amplification [53]. Bartunik et al. [54] utilized the concentration of biocompatible magnetic nanoparticles for the encoding of information with a receiver detector based on susceptibility [54].

Angerbauer et al. [18] developed an experimental microfluidic MC testbed based on the salinity of the fluid. The information is encoded in multiple levels of salt concentration and the receiver measures conductivity with electrodes. Specifically, in this experimental testbed, an integrated mixing structure in the channel mixes the injected salt solution and the background flow of distilled water [18]. We utilized the experimental testbed of Angerbauer et al. [18] to benchmark our OpenFOAM simulation results.

III. SCENARIO AND PRELIMINARIES

This section describes the scenario for the IoBNT, the system model, and preliminaries for the simulation of microfluidic MC.

A. IoBNT SCENARIO

In our scenario, we explore the application of microfluidic MC within the context of the IoBNT, e.g., for targeted drug delivery, with the model illustrated in Fig. 1. In our simulations, at the transmitter, also called inlet, we embed a nanoscale device within a microfluidic channel. This device is designed to release specific particles into the microfluidic channel in a controlled and uniform manner. The particles represent encoded information, which needs to be transmitted to the receiver. Thereby, the transmitter is programmed to release the particles uniformly across the microfluidic

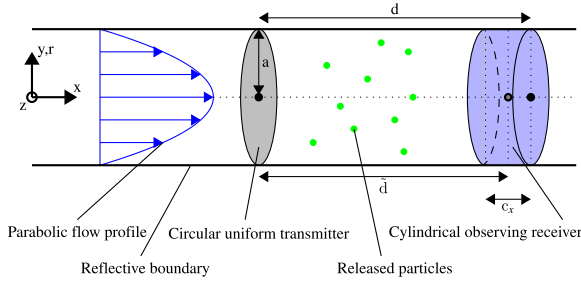


FIGURE 1. Channel model for a circular uniform transmitter and a cylindrical observing receiver, following [22, Fig. 1]. Thereby, the observing receiver is transparent, i.e., does not impact the transport of the particles [16]. Furthermore, the receiver detects and counts the number of particles within its receiver volume.

channel's cross-section. The microfluidic channel serves as the medium through which the particles travel. At the receiver, later in simulations called outlet, there is another nanoscale device within the microfluidic channel. This device is equipped with sensors capable of detecting the released and transmitted particles. The receiver's role is to capture the molecular signals, decode them, and interpret the transmitted information. The receiver's sensors are strategically placed across the cross-section of the microfluidic channel to sense all the passing particles, i.e., to detect and count the number of the particles within the receiver's volume.

B. MC SYSTEM MODEL

For comparability, we base our system model on the model proposed in [16]. We examine a cylindrical duct that is impermeable with an infinite length, i.e., axial extent, and radius a [16]. The duct can be represented using cylindrical coordinates (x, r, φ) , where $x \in (-\infty, \infty)$ denotes the axial position, $r \in [0, a]$ represents the radial distance from the center of the duct, and $\varphi \in (-\pi, \pi]$ covers the azimuth angle [16]. The fluid inside the duct possesses a dynamic viscosity denoted by η and undergoes steady laminar flow in the x -direction. Based on the ratio of inertial to viscous forces [55], known as the *Reynolds* number, laminar and turbulent flows can be distinguished. In a pipe or a tube, the *Reynolds* number can be calculated as [56, Eq. (5.2)]:

$$Re = \frac{v_{\text{eff}} D_H}{\nu}, \quad (1)$$

whereby v_{eff} denotes the channel's mean velocity, D_H is the hydraulic diameter of the tube (the hydraulic diameter for a circular tube is equal to the diameter of the tube), and ν is the kinematic viscosity.

At the transmitter, OOK modulation is employed, whereby the symbol time interval length is denoted as T corresponding to a data rate of $1/T$. To transmit a bit-1 or bit-0 during the k -th symbol interval, the transmitter releases N_{TX} or 0 particles, respectively, at time kT , $k = 0, 1, 2, \dots$. For the particle release, we consider the following scenario: The particle distribution across the cross-section at $x = 0$ is uniform and random [16]. More specifically, the circular uniform

transmitter, which is visualized in gray color in Fig. 1, releases these particles at $x = 0$. The particles (green colored dots in Fig. 1) propagate from the transmitter side to the cylindrical receiver, mainly governed by advection due to the flow in the channel (for the flow-dominated regime, see Section IV). A reflective boundary at the duct wall reflects the particles in the channel, i.e., the particles do not pass through the duct wall and are not absorbed by the duct wall.

The cylindrical receiver is transparent, i.e., it does not impact the particle transport [16]. The cylindrical receiver is positioned at an axial transmitter-to-mid-receiver distance \tilde{d} with a length, i.e., axial extent (length of the receiver in the x -direction) c_x , see Fig. 1. Also, d denotes the distance between the transmitter and the end of the sensing volume of the receiver, here called transmitter-to-end-receiver distance. The cylindrical receiver is capable of detecting and counting particles within its volume for sensing, i.e., the cylindrical volume $\pi a^2 c_x$ encompassing the full radius a of the duct.

Following [16], the detection and demodulation of the received binary OOK symbols $\tilde{b}[k] \in \{0, 1\}$ is performed based on the number of observed particles $N_{\text{RX}}(t)$, and a pre-defined decision threshold $\xi \in \{0, 1, \dots, N_{\text{TX}}\}$ [16, Eq. (1)]:

$$\tilde{b}[k] = \begin{cases} 0, & \text{if } N_{\text{RX}}(t_0 + kT) < \xi \\ 1, & \text{if } N_{\text{RX}}(t_0 + kT) \geq \xi, \end{cases} \quad (2)$$

whereby t_0 and $\tilde{b}[k]$ denote the detection delay and the OOK symbol detected in the k -th symbol interval. The released particles undergo transportation through the fluid flow and Brownian motion. Furthermore, analogous to [16], we consider a scenario where the particles do not interact with each other and have no impact on the flow field [57]. Due to their small size, other forces, such as gravity, can be neglected [16].

C. MC FUNDAMENTALS

This section briefly reviews the fundamentals of MC which involves the transmission of information through mass transfer [16], [58]. The mass transfer within fluids is facilitated by both fluid flow and Brownian motion, known as advection and diffusion, respectively [16]. Consequently, the process of the mass transfer can be defined by a spatial Probability Distribution Function (PDF) $p(\mathbf{r}, t)$ which denotes a normalized concentration. Thereby, the term $dVp(\mathbf{r}, t)$ signifies the fraction of particles on average within the differential volume dV at the location \mathbf{r} and time t . The PDF $p(\mathbf{r}, t)$ can be determined as the solution to a specific partial differential equation, known as the advection-diffusion equation [57, Eq. (5.22)]:

$$\frac{\partial}{\partial t} p = D \nabla^2 p - \nabla p \mathbf{v}(\mathbf{r}), \quad (3)$$

where ∇ denotes the Nabla operator, D is the diffusion coefficient, and $\mathbf{v}(\mathbf{r})$ is the vector of the velocity at the point \mathbf{r} . For solving Eq. (3), the velocity field $\mathbf{v}(\mathbf{r})$ has to be

known [16]. Following [57, Ch. 2], the *Navier-Stokes* equation describes fundamentally the fluid flow, see Section V, which enables the evaluation of the velocity field. Dealing with straight and rigid channels [16] characterized by no-slip boundary conditions—the velocity at the boundary is equal to zero—and subjected to pressure-driven flow in a steady-state scenario, the resulting velocity distribution is known as *Poiseuille* flow [57, Ch. 3]. Assuming that the fluid follows *Newtonian* behavior, where its properties can be described by dynamic viscosity η , the following equation follows for two dimensions [57, Eq. (3.32)]:

$$v(r) = 2 v_{\text{eff}} \left(1 - \frac{r^2}{a^2} \right), \quad (4)$$

with v_{eff} as the mean velocity [57, Eq. (3.34)]:

$$v_{\text{eff}} = |\partial_x P| \frac{a^2}{8\eta}, \quad (5)$$

whereby $\partial_x P$ denotes the gradient of the pressure. The maximum velocity v_{max} occurs at the center, i.e., $y = 0$, following [16, Fig. 1], or using $r = 0$ in Eq. (5).

Furthermore, the *Péclet* number Pe characterizes the proportional significance of advection and diffusion [16], mathematically [59, Eq. (4.6.8)]:

$$\text{Pe} = \frac{v_{\text{eff}} a}{D}. \quad (6)$$

IV. ANALYTICAL APPROACH OF THE MICROFLUIDIC CHANNEL

This section briefly reviews the analytical fundamentals required for the closed-form, analytical solution of a uniform release of particles at the transmitter side. We include this review of the theoretical channel modeling in order to provide background for generalist readers of this methods article. We will compare our OpenFOAM MPPIC solver simulation results with the analytical results obtained from the reviewed analytical modeling. Due to the independence of the velocity field of the axial position, Eq. (3) can be simplified for $t > 0$ [16, Eq. (5)]:

$$\frac{\partial}{\partial t} p = D \nabla^2 p - v(r) \frac{\partial}{\partial x} p. \quad (7)$$

Thereby, we assume $\frac{\partial}{\partial r} p = 0$ at the boundary $r = a$, $p(x, r, \varphi; 0) = \delta(x)/(\pi a^2)$ for a uniform release [16], as considered in this paper (cf. Section III-B). Following [16, Sec. 3], Eq. (7) can be solved in closed-form for certain parameter regimes. Wicke et al. [16, Eq. (6)] seek the probability for a time-dependent observation, in the following also called CIR of the MC channel:

$$P_{\text{ob}}(t) = \int_{V_{\text{RX}}} p(x, r, \varphi; t) dV_{\text{RX}}, \quad (8)$$

where V_{RX} denotes the volume of the receiver.

For an analytical solution, there are two regimes, the dispersion regime and the flow-dominated

TABLE 1. Parameter settings of the analytical channel model [16].

Parameter	Symbol	Value	Unit
Transmitter-to-end-receiver distance	d	{200; 800; 1600}	μm
Transmitter-to-mid-receiver distance	\tilde{d}	{150; 750; 1550}	μm
Radius	a	200	μm
Receiver length	$c_x = a/2$	100	μm

regime [59, Ch. 4.6].

$$\frac{v_{\text{eff}} a}{D} \ll \frac{4\tilde{d}}{a} \quad \text{for dispersion regime} \quad (9)$$

$$\frac{v_{\text{eff}} a}{D} \gg \frac{4\tilde{d}}{a} \quad \text{for flow-dominated regime.} \quad (10)$$

Following the parameters introduced in Section VI-A1, a flow-dominated regime follows, see Eq. (11) and [16, Fig. 2]. For the channel model properties in Table 1, and specifically, for $\tilde{d} = 150 \mu\text{m}$,

$$\frac{v_{\text{eff}} a}{D} = 200 \gg 3 = \frac{4\tilde{d}}{a}; \quad (11)$$

while for $\tilde{d} = 750 \mu\text{m}$, $4\tilde{d}/a = 15$, and for $\tilde{d} = 1550 \mu\text{m}$, $4\tilde{d}/a = 31$. It should be noted that the transmitter-to-mid-receiver distance \tilde{d} depends on the selected length c_x of the cylindrical receiver, i.e., $\tilde{d} = d - c_x/2$.

Assuming a uniform release of the particles at $x = 0$, all particles will arrange themselves upon a paraboloid surface [16], progressively extending along the axis as time elapses, displaying symmetrical rotation [16]. Consequently, the distribution is confined to the cross-sectional plane and is delineated by the r -coordinates and φ -coordinates, which preserves its uniformity due to the flow occurring exclusively in the x -direction [16]. With $v(r)$ following Eq. (4), the spatial distribution can be expressed as [16, Eq. (15)]:

$$p(x, r, \varphi; t) = \frac{1}{\pi a^2} \delta(x - v(r)t). \quad (12)$$

Integrating over the volume of the receiver, the analytical CIR CIR_{ana} in a flow-dominated regime for a uniform release follows [16, Eq. (16)]:

$$\text{CIR}_{\text{ana}}(t) = \begin{cases} 0, & \text{if } t \leq t_1 \\ 1 - \frac{\tilde{d} - c_x/2}{2v_{\text{eff}} t}, & \text{if } t_1 \leq t \leq t_2 \\ \frac{c_x}{2v_{\text{eff}} t}, & \text{if } t \geq t_2. \end{cases} \quad (13)$$

The time instant t_1 can be calculated as [16, Eq. (17)]:

$$t_1 = \frac{\tilde{d} - c_x/2}{2v_{\text{eff}}}. \quad (14)$$

The time instant t_2 , which attains the maximum of CIR_{ana} , is obtained by replacing the numerator in Eq. (14) by $\tilde{d} + c_x/2$.

V. SIMULATION BASICS OF THE MICROFLUIDIC CHANNEL IN OPENFOAM

This section reviews the basics of the simulation of MC in a microfluidic channel using OpenFOAM. Within the OpenFOAM framework, the Finite Volume Method (FVM) is employed to spatially discretize the partial differential equations. This involves dividing the continuous medium into control volumes, wherein the equation is formulated as a volume integral over each control volume. The majority of CFD solvers in OpenFOAM either use the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) or PISO algorithm [60]. The utilization of the SIMPLE algorithm enables an extension of the time interval, leading to enhanced convergence due to the implicit treatment of viscous and convection terms [60]; however, the SIMPLE algorithm incurs high computational complexity for unsteady or turbulent flows. In contrast, the PISO algorithm, introduced by Issa [48], introduces a separation between the pressure and the velocity, thereby avoiding the iterative computations that are intrinsic to the SIMPLE algorithm. This PISO approach proves advantageous, particularly for unsteady flow scenarios, facilitating the accurate computation of unsteady flows with a limited number of iterations. We adopt the Pressure-Implicit with Splitting of Operators for Pressure-Linked Equations (PIMPLE) algorithm for numerical simulations of the fluid phase. The PIMPLE algorithm amalgamates the benefits of both the PISO and SIMPLE algorithms, combining their strengths to achieve accurate simulations for unsteady flows while mitigating excessive computational demands [60].

A. SOLVER

A transient two-phase solver, the MPPIC solver, was proposed by Andrews and O'Rourke in 1996 [17] to simulate "the coupled transport of a single kinematic particle cloud" [61], wherein the influence of the particles' volume fraction on the continuous phase is considered. This modeling approach utilizes the MPPIC method to account for collisions without explicitly resolving the particle-particle interactions. The numerical MPPIC method models fluid-solid and solid-solid interactions [60]. In 2001, Snider [62] extended the MPPIC method to three dimensions.

1) FLUID PHASE

Numerical simulations in this study are performed under isothermal conditions, and the fluid acts as an incompressible *Newtonian* fluid. For a given fluid volume, the continuity equation is defined as [60, Eq. (1)]:

$$\frac{\partial}{\partial t}(\epsilon_f \rho_f) + \nabla(\epsilon_f \rho_f \mathbf{u}_f) = 0, \quad (15)$$

whereby t denotes the time, ϵ_f is the fraction of the fluid phase, ρ_f is the fluid density, \mathbf{u}_f is the fluid velocity vector, and ∇ is the Nabla operator. Following Eq. (15), the

momentum equation for the fluid is defined as [60, Eq. (2)]:

$$\begin{aligned} \frac{\partial}{\partial t}(\epsilon_f \rho_f \mathbf{u}_f) + \nabla(\epsilon_f \rho_f \mathbf{u}_f \mathbf{u}_f) \\ = -\epsilon_f \nabla p_f + \nabla(\epsilon_f \boldsymbol{\tau}_f) + \epsilon_f \rho_f \mathbf{g} - \mathbf{F}_{fs}, \end{aligned} \quad (16)$$

whereby p_f denotes the fluid pressure, \mathbf{g} is the acceleration due to gravitation, and \mathbf{F}_{fs} is the exchange of the momentum between the fluid and the solid. Also, $\boldsymbol{\tau}_f$ denotes the stress tensor of the fluid and is defined as [60, Eq. (3)]:

$$\boldsymbol{\tau}_f = \mu_f [\nabla \mathbf{u}_f + (\nabla \mathbf{u}_f)^T] - \frac{2}{3} \mu_f \nabla \mathbf{u}_f \mathbf{I}, \quad (17)$$

with μ_f denoting the fluid viscosity and \mathbf{I} as the unit tensor.

2) SOLID PHASE

The solid particles are modeled as spheres, and multiple particles are tracked as a unit parcel. The PDF $f_s(\mathbf{x}_s, \mathbf{v}_s, m_s, t)$ describes the dynamics of the particles, whereby \mathbf{x}_s denotes the position of the particle, \mathbf{v}_s is the velocity, m_s is the mass, and t denotes the time instant. The transport equation for f_p is [60, Eq. (4)]:

$$\frac{\partial f_s}{\partial t} + \frac{\partial (f_s \mathbf{v}_s)}{\partial x} + \frac{\partial (f_s \mathbf{A}_s)}{\partial \mathbf{v}_s} = \frac{f_D - f_s}{\tau_D}, \quad (18)$$

whereby f_D denotes the local mass-averaged particle velocity [63] and τ_D is the collision damping time [64]. The particle acceleration \mathbf{A}_s follows [60, Eq. (5)]:

$$\mathbf{A}_s = \frac{d\mathbf{v}_s}{dt} = D_s(\mathbf{v}_f - \mathbf{v}_s) - \frac{\nabla p_f}{\rho_f} + \mathbf{g} - \frac{\nabla \tau_s}{\epsilon_s \rho_s} + \frac{\bar{\mathbf{v}}_s - \mathbf{v}_s}{2\tau_D}. \quad (19)$$

The fluid velocity is denoted by \mathbf{v}_f , the interphase momentum transfer coefficient by D_s , while ρ_s is the density of the solid (particles). Furthermore, τ_s denotes the normal stress of the particle contact, ϵ_s is the solid volume fraction, and $\bar{\mathbf{v}}_s$ is the local mass-averaged velocity of the solid. As a model for the collision between the particles, the *Harris and Crighton* model [65] is applied. Thereby, the normal stress of the particle contact τ_s is defined as [60, Eq. (6)]:

$$\tau_s = \frac{P_s \epsilon_s^\beta}{\max[\epsilon_{cp} - \epsilon_s, \alpha(1 - \epsilon_s)]}, \quad (20)$$

whereby P_s denotes the particle pressure, β is a non-dimensional model parameter, ϵ_s is the local maximum volume fraction, ϵ_{cp} is the volume fraction in close-pack, and α is a constant avoiding singularity.

3) COUPLING THE PHASES

For coupling the phases, the following equation computes the momentum exchange F_{fs} between the fluid and the solid phases [60]:

$$\begin{aligned} F_{fs} = - \int \int f m_s [D_s(\mathbf{v}_f - \mathbf{v}_s) - \frac{1}{\rho_s} \nabla p + \mathbf{v}_s \frac{dm_s}{dt}] \\ \times dm_s d\mathbf{v}_s dT_s, \end{aligned} \quad (21)$$

whereby f denotes the particle PDF, m_s is the mass of the particle, and T_s is the temperature of the particle.

TABLE 2. Meshing parameters of the different meshes created with *Salome*.

Duct model	# of nodes	# of control volumes
$a = 200 \mu\text{m}$ & $d = 200 \mu\text{m}$	21074	106608
$a = 200 \mu\text{m}$ & $d = 800 \mu\text{m}$	74544	395343
$a = 200 \mu\text{m}$ & $d = 1600 \mu\text{m}$	142944	764437

VI. METHODOLOGY FOR MICROFLUIDIC MC USING OPENFOAM

This section describes all steps of the pre-processing, the simulation parameters, and finally, the hard- and software parameters used for obtaining the simulation results. We make the entire source code and simulation results publicly available [19]. We simulated the cases in Table 2 with a *Newtonian* fluid, namely water, and compared with the analytical approximations in [16] and PBSs.

A. SIMULATION SETTINGS AND PARAMETERS

1) GEOMETRY AND MESHING

To evaluate the application of OpenFOAM, and especially the MPPIC solver, as a simulation tool for microfluidic MC, we consider a straight impermeable cylindrical duct [16]. Thereby, to make the simulation results comparable to the analytical results from Wicke et al. [16], we use the same physical parameter values as in [16]. Specifically, we consider a diffusion coefficient of $D = 1 \times 10^{-10} \text{ m}^2 \text{ s}^{-1}$, reasonably approximating small proteins [16], [66]. Furthermore, we consider the duct radius $r = 200 \mu\text{m}$, which is a reasonable approximation for ducts in the microfluidic domain [57]. We initially considered the two transmitter-to-mid-receiver distances (transmitter-to-end-receiver distances) $\tilde{d} = 150 \mu\text{m}$ ($d = 200 \mu\text{m}$) and $\tilde{d} = 750 \mu\text{m}$ ($d = 800 \mu\text{m}$) [16]. Based on the initial simulation results for these two distances, we added the distance $\tilde{d} = 1550 \mu\text{m}$ ($d = 1600 \mu\text{m}$). Furthermore, to compare the simulation results to the analytical results, we consider the receiver length $c_x = a/2$ for a full channel, cylindrical, observing receiver from Section III; see also Table 1.

We created the geometry using the open-source software *Salome*, where the models have also meshed with the parameters in Table 2. For each created model, the global maximum edge length of a control volume is $10 \mu\text{m}$ and the minimum edge length of a control volume is $1 \mu\text{m}$.

2) FLUID, WALL, AND PARTICLE PROPERTIES

Table 3 gives an overview of the parameters of the solid and fluid phases.

Water is used as an incompressible, *Newtonian* fluid in the simulations. Thereby, we assume a density of $\rho_{\text{water}} = 1000 \text{ kg m}^{-3}$ and a kinematic viscosity of $\nu = 1 \times 10^{-6} \text{ m}^2 \text{ s}^{-1}$ (for the temperature $T = 20^\circ\text{C}$) [69]. In addition, we assume a scenario in which the particles are released, and the flow in the system is already present, i.e., steady-state flow is predominant since this is the only

TABLE 3. Simulation parameters for the microfluidic Molecular Communication (MC) tube.

Parameter	Value
Fluid phase	
Temperature [K]	293.15
Density water [kg/m^3]	1000.0
Dynamic viscosity [mm^2/s]	1.0
Drag model	Ergun Wen-Yu [67], [68]
Solid phase	
Particle diameter [nm]	5.0
Particle density [kg/m^3]	1335.0
Close packing volume fraction	0.58
Total number of released parcels	998.6 ± 13.4 ($d = 200 \mu\text{m}$), 998.3 ± 13.4 ($d = 800 \mu\text{m}$), 999 ± 0 ($d = 1600 \mu\text{m}$)
Number of particles per parcel	1

way to obtain simulation results that are comprehensible, reproducible, and comparable with analytical methods.

Steady-state flows exhibit time independence, i.e., the gradient with respect to the time is zero. The flow velocity u in the main flow direction, here x , is the same at any time t , i.e., $\partial u / \partial t = 0$. To determine the time frame for settling into a quasi-steady state, see also [70], we performed simulations without injecting any particles for the duct models in Table 2. Reaching a steady state in CFD simulations ensures the reliability and accuracy of the simulation results [71, Ch. 5.13], [72, Ch. 7]. The steady state represents a condition where all transient effects have dissipated, allowing the flow variables to become time-independent, which is essential for comparing with experimental measurements. In Fig. 2, we plot the velocity component in the x -direction u at the outlet over time. Fig. 2 demonstrates the convergence to a steady state from about 0.4 seconds after the start of the simulation. Fig. 2 also demonstrates pronounced velocity changes at the outlet within the first 0.4 s after the simulation start for the three considered transmitter-to-end-receiver distances d . Consequently, we set a Start of Injection (SOI) of 0.4 s for injecting the particles for the simulations.

Furthermore, we consider rigid rough walls. For the mean velocity in the channel $v_{\text{eff}} = 1 \text{ mm s}^{-1}$, the hydraulic diameters of the pipe $D_H = 2 \times a = 400 \mu\text{m}$, and the kinematic viscosity $\nu = 1.0 \text{ mm}^2 \text{ s}^{-1}$, the *Reynolds* number follows from Eq. (1):

$$Re = \frac{v_{\text{eff}} D_H}{\nu} = 0.4 \ll 2100. \quad (22)$$

The *Reynolds* number of 2100 [73] denotes the transition from laminar to turbulent flow in pipes. Consequently, laminar flow is present in our model for the CFD simulation. In the microfluidic environment, laminar flow is typically assumed [23], [57].

The released particles align in size and density with hemoglobin, a protein complex found as a blood pigment in human red blood cells that transports oxygen in the bloodstream. For the simulation, $N_{\text{TX}} = 10^3$ particles are released at the inlet at the time $t = \text{SOI} = 0.4 \text{ s}$. The particles are randomly and equally distributed over the inlet surface. Following [74, Table 4], we assume an average particle size of

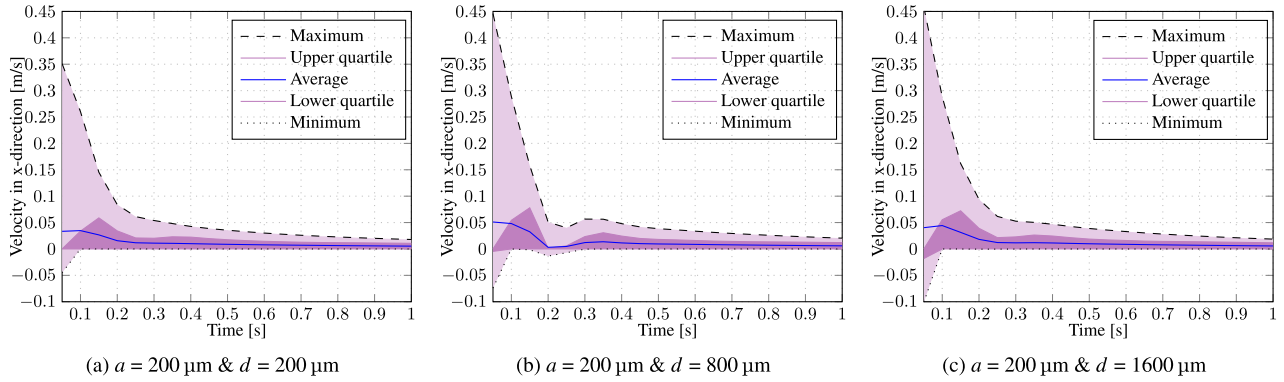


FIGURE 2. Convergence to steady state for different transmitter-to-end-receiver distances d . The simulations were performed with the *simpleFoam* solver, i.e., without interaction with injected particles. A quasi-stationary state is established in all three cases from approximately 0.4 seconds onwards; therefore, we set the Start of Injection (SOI) to 0.4 seconds.

TABLE 4. Boundary conditions of the simulation cases for the velocity and pressure at the inlet, outlet, and walls; whereby, the $[x, y, z]$ notation indicates the parameter values for the three spatial dimensions, and “uniform 0” denotes a vector consisting of three zeros.

Boundary	Patch	Type	Value
Velocity [m/s]	inlet	<i>fixedValue</i>	$[0.001, 0, 0]$
	outlet	<i>inletOutlet</i>	$[0, 0, 0]$
	walls	<i>noSlip</i>	–
Kinematic pressure $[m^2/s^2]$	inlet	<i>fixedFluxPressure</i>	uniform 0
	outlet	<i>fixedValue</i>	uniform 0
	walls	<i>fixedFluxPressure</i>	uniform 0

5 nm, and a particle density of 1.335 g cm^{-3} [75]. In order to achieve an immediate release of 1000 particles when the time instant $\text{SOI} = 0.4 \text{ s}$ is reached, 1000000 packages per second are released for a duration of 0.001 s. After evaluation of all $n = 30$ simulation runs ($n = 10$ simulation runs per case), the average over all simulations is a total number of 998.6 ± 13.4 ($d = 200 \mu\text{m}$), 998.3 ± 13.4 ($d = 800 \mu\text{m}$), and 999 ± 0 ($d = 1600 \mu\text{m}$) released particles (at one particle per packet) for the three simulation cases. The slight deviations result in slightly different time steps for each calculation when the parameter “AdjustTimeStep” is enabled.

3) BOUNDARY CONDITIONS

Table 4 summarizes the boundary conditions at the inlet, the outlet, and the walls.

At the inlet, the type *fixedValue* was chosen for the boundary condition of the velocity. The face values are [76]:

$$\phi_f = \phi_{\text{ref}}, \quad (23)$$

where ϕ_f denotes the face value and ϕ_{ref} denotes the reference value of the velocity. Thereby, the vector $[0.001, 0, 0]$ defines the velocity present at the face, here inlet, in all three spatial directions. We assume a flow rate of the fluid of 1 mm s^{-1} so that the first component of the vector is set equal to 0.001 m s^{-1} . On the walls, we set the boundary condition *noSlip*, which reduces the velocity at the patch to zero in all three spatial directions [77], i.e., $\phi_{\text{wall}} = [0, 0, 0]$.

Finally, the *inletOutlet* boundary condition at the outlet sets the system’s outflow as a zero gradient condition. The zero gradient condition sets the field to the internal field value so that $\partial\phi/\partial n = 0$ [76].

For the pressure boundary condition at the inlet and the walls, the *fixedFluxPressure* condition was applied. This particular boundary condition is applied in pressure scenarios, where the zero gradient condition is typically adopted. The application of the *fixedFluxPressure* condition becomes pertinent in situations where external forces, e.g., gravity and surface tension, are influential components within the solution equations [61]. The *fixedFluxPressure* condition effectively adapts the gradient to accommodate these factors [61]. At the outlet, the *fixedValue* condition is used, see Eq. (23).

B. COMPUTATION CONTROL PROPERTIES

The simulation runs in OpenFOAM were carried out for all three cases (combinations of a and d values) with the enabled parameter “AdjustTimeStep”. When the parameter “AdjustTimeStep” is enabled, the time step is adapted based on the simulation’s maximum *Courant* number Co , here chosen as 0.9. The *Courant* number is defined as [78]:

$$\text{Co} = \sum_i \frac{v_i \Delta t}{\Delta h_i}, \quad (24)$$

whereby i denotes the number of dimensions, i.e., here $i = 3$, v_i is the flow velocity of the fluid in dimension i , Δt is the time step duration, and Δh_i is the characteristic size of the mesh control volume in dimension i . A *Courant* number larger than one indicates that information may propagate across more than one control volume during a single time step [78]. This can result in inaccuracies in the solution and may potentially lead to nonphysical outcomes or divergence in integration schemes.

The enabled parameter “AdjustTimeStep” adjusts the time step size dynamically during each simulation run to avoid these inaccuracies and nonphysical outcomes, which could lead to the computation job being canceled. Typically,

the enabled parameter “AdjustTimeStep” decreases the computation time significantly compared to a constant time step size. A disadvantage of the enabled “AdjustTimeStep” is that the simulation results vary slightly for each time step output in a given simulation run. In order to achieve reproducible simulation results—in addition to the slightly varying simulation results with the enabled parameter “AdjustTimeStep” (for which we ran $k = 10$ simulation runs per case)—we also conducted additional simulation runs with the disabled parameter “AdjustTimeStep”. Independent simulation runs for a given scenario with the disabled parameter “AdjustTimeStep” (and a fixed number of allocated computing processor cores) gave exactly the same results.

C. HARD- AND SOFTWARE PROPERTIES

As already described in Section VI-A1, the *Salome* tool is used to create the geometry and the mesh. The computation of the simulation is performed on a High Performance Computer (HPC) system. For this purpose, the calculations are parallelized and performed on a total number of 50, 100, or 120 subdomains. Each subdomain represents a processor core. An allocation of the subdomains takes place by allocation to the Intel Haswell CPUs, consisting of among others of 594 nodes with 2.67 GB RAM per core. The OpenFOAM version used is OpenFOAM v2212 for local calculations and OpenFOAM v2206 for calculations on the HPC. Finally, for post-processing, we used the tool *ParaView*, an open-source application for scientific visualization. A *Python* program [19] was coded to analyze the OpenFOAM folder structure for each simulated time step with respect to the CIR.

D. PARTICLE-BASED SIMULATION

Simulation results obtained in OpenFOAM are compared with PBS [38] results for a 3-D *Poiseuille* flow. In the PBS [38] in *Matlab*, at the time instant $t = 0$, the time of injection, $N_{TX} = 1000$ particles are released uniformly distributed at the planar transmitter. The simulation step time is $\Delta t_{Sim} = 0.02$ s, as the written step time in the OpenFOAM simulation. The other input parameters, i.e., the radius of the tube r , the transmitter-to-end-receiver distance d , the diffusion coefficient D , and the average velocity of the fluid v_{eff} have been adopted analogously to the values for the OpenFOAM simulation, see Section VI-A. To determine the number of particles in the cylindrical volume of the receiver, the position of each particle in each time step is evaluated in the post-processing, i.e., the number of the particles in the sensitive volume of the receiver can be determined. The OpenFOAM simulations follow the same procedure, i.e., the simulations are carried out for all transmitter-to-end-receiver distances d , and the number of particles in the volume $\tilde{d} - c_x/2 \leq x \leq \tilde{d} + c_x/2 = d$ of the cylindrical receiver is determined in post-processing. The *Python* program for post-processing is also available in the repository [19].

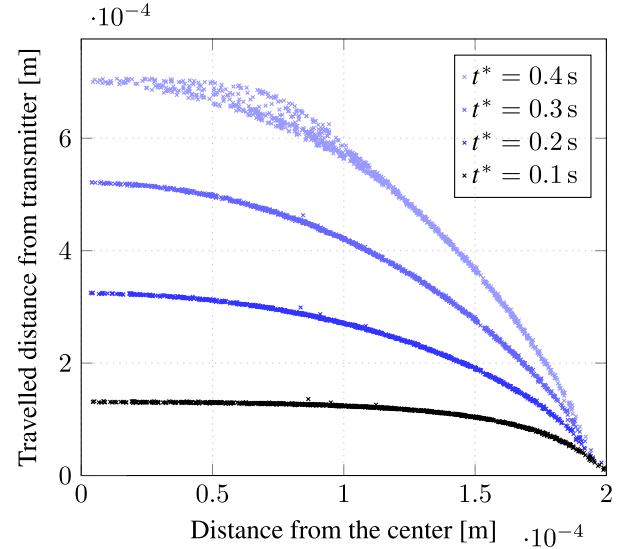


FIGURE 3. Formation of the parabolic flow profile in the fluid volume for $a = 200 \mu\text{m}$ & $d = 800 \mu\text{m}$ for one simulation run in OpenFOAM. Each data point identifies one particle. t^* denotes the time after the Start of Injection (SOI) (0.4 s), e.g., $t^* = 0.1$ s refers to $T = 0.5$ s simulation time.

VII. SIMULATION RESULTS

This section examines the formation of the parabolic flow profile as well as the CIR of the simulated systems. Also, we evaluate and explain the differences between the analytical approach, the OpenFOAM simulation results, and the PBS results.

Fig. 3 shows exemplarily for the selected geometry ($a = 200 \mu\text{m}$ & $d = 800 \mu\text{m}$) the spreading of the particles for different time steps in the OpenFOAM simulation. The horizontal axis shows a particle’s distance from the center of the tube (where $r = 0$). The maximum distance is $200 \mu\text{m}$, corresponding to the radius a of the cross-section. In Fig. 3, all released particles were considered. The vertical axis shows the distance of a particle traveled from the transmitter, i.e., the point of release. As time increases, the particles move in the channel; consequently, the traveled distance increases. Furthermore, Fig. 3 indicates the formation of the parabolic flow profile: the particles at the center of the flow, i.e., at $r = 0$, move with the highest velocity and therefore cover the greatest distance. Whereas, the particles near the edge, i.e., near $r = a = 200 \mu\text{m}$, move with very low velocity and therefore cover a shorter distance.

A. CHANNEL IMPULSE RESPONSE (CIR)

To evaluate the CIR in OpenFOAM, the *kinematicCloud-OutputProperties* can be used for each time step. For each time step i , the fraction of received particles $F_{\text{parcel},i}$ can be evaluated as:

$$F_{\text{parcel},i} = \frac{\sum_{i=1}^m n_{\text{received},i} - \sum_{i=1}^{m-1} n_{\text{received},i}}{\sum_{i=1}^m n_{\text{released},i}}, \quad (25)$$

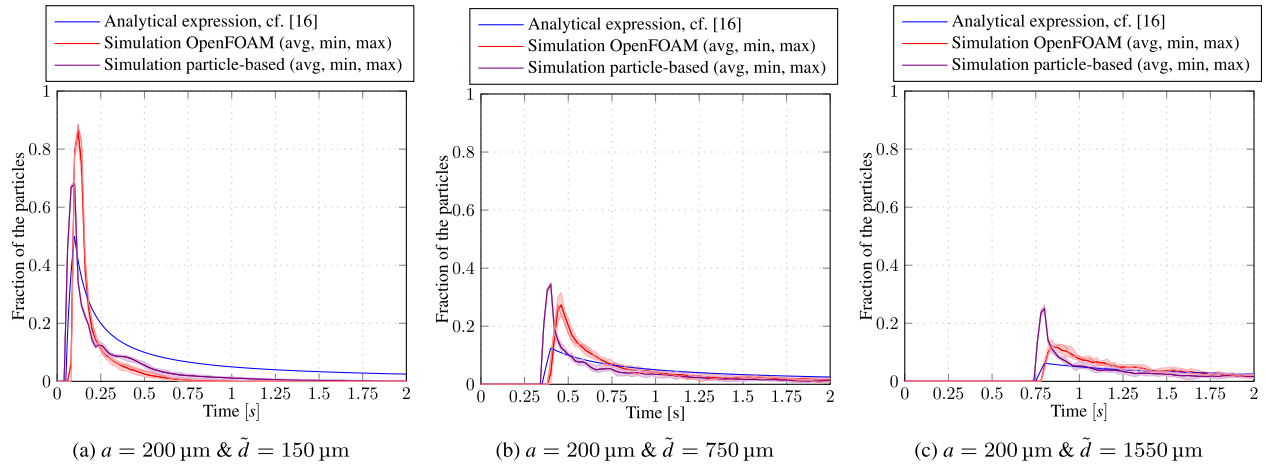


FIGURE 4. Channel Impulse Response (CIR): Comparison of the fractions of received particles for the simulation results for different transmitter-to-end-receiver distances \tilde{d} obtained from OpenFOAM, the Particle-Based Simulation (PBS), and the analytical approach in [16] as a function of the elapsed time since the release of the particles at Start of Injection (SOI). For graphical representation, we colored the area between the maximum and minimum values for each time step in red for OpenFOAM and violet for PBS.

whereby $\sum_{i=1}^m n_{\text{received},i}$ denotes the number of received particles up to and including the time step m , $\sum_{i=1}^{m-1} n_{\text{received},i}$ is the number of received particles up to and including the previous time step $m-1$, and $\sum_{i=1}^m n_{\text{released},i}$ is the number of released particles over all time steps ($i = 1$ to $i = m$).

Fig. 4 compares the analytical approach with the results of the computational simulations in OpenFOAM and the PBSs for the three duct models in Table 2, i.e., $\tilde{d} = 150 \mu\text{m}$, $\tilde{d} = 750 \mu\text{m}$, and $\tilde{d} = 1550 \mu\text{m}$. Thereby, the lower and upper constraints, i.e., the minima and maxima for each plotted time step 0 s, 0.02 s, 0.04 s, ..., 2 s do not represent a single simulation run, but rather visualize the minimum and maximum value for each time step over all simulation runs (envelope curve).

Fig. 4 indicates that the simulated fractions of particles after the release at the SOI exhibit a rapid increase that nearly coincides with the fractions of particles obtained from the analytical expression. Subsequently, the simulated fractions of particles tend to decline faster than the analysis. However, the greater the transmitter-to-mid-receiver distance \tilde{d} , the better the correspondence between the simulation results and the analytical expression.

The simulated fractions of particles in Fig. 4 exhibit slight variations that are superimposed on the declining curves. The PBS involves random numbers for simulating the particle movements, which likely contribute to the observed variations. We further investigate the impact of the setting of the “AdjustTimeStep” parameter in OpenFOAM (which should be deterministic in nature due to the solution of a meshed differential equation) and the allocation of the processing cores in the HPC on the variations. We ran all three simulation cases without the enabled parameter “AdjustTimeStep,” i.e., with a constant time step of $\Delta t = 1 \times 10^{-5}$ s, and on a constant number of cores,

i.e., 4 cores, and plot these OpenFOAM simulation results in Fig. 4 in light red line style (in contrast to the averaged smoothed dark red curve for the simulation with enabled “AdjustTimeStep”). As Fig. 4 indicates, the OpenFOAM simulations results without enabled parameter “AdjustTimeStep” and with a constant number of cores are perfectly reproducible, i.e., these OpenFOAM simulation results match exactly from one independent simulation run (replication) to the next. We also observe that these perfectly reproducible OpenFOAM simulation results essentially overlap with the OpenFOAM simulation results with the enabled parameter “AdjustTimeStep” and with a varying allocation of processor cores. Specifically, comparing the reproducible OpenFOAM simulation results with the OpenFOAM simulation results with enabled parameter “AdjustTimeStep”, the Root-Mean-Square Error (RMSE) values for $\tilde{d} = 150 \mu\text{m}$, $\tilde{d} = 750 \mu\text{m}$, and $\tilde{d} = 1550 \mu\text{m}$ are 1.43 %, 1.29 %, and 0.55 % of the absolute fraction of received particles in a time slot, respectively.

Thus, we can overall conclude that the variations due to the “AdjustTimeStep” and variable HPC core allocation are minuscule and that for experimental convenience and time savings it is generally prudent to run the OpenFOAM simulations with the “AdjustTimeStep” and the HPC core allocation. In particular, without enabling “AdjustTimeStep,” one always has to specify the time step duration for each simulation time step. The enabled “AdjustTimeStep” adapts the time step duration dynamically based on the mesh coarseness during the simulation run, i.e., the simulation run will typically finish earlier. Also, specifying the time step duration has been relatively simple for our simulation cases due to the constant control volume size. For non-constant control volume sizes, e.g., for a finer resolution near the walls for turbulence modeling, the computation times significantly increase without “AdjustTimeStep.” In particular, the enabled “AdjustTimeStep” can adaptively set these short time step durations when they are needed but otherwise

operate with longer (coarser) time steps; without enabled “AdjustTimeStep,” the entire simulation would need to run with the short time step durations (and would therefore take longer overall).

The peak of each case has a slightly higher magnitude value for the OpenFOAM simulation results than for the analytical approach. However, the greater the transmitter-to-mid-receiver distance \tilde{d} , the lower the difference in the peak magnitude values of the fraction of the particles of the simulation vs. analysis. While the difference in the peak values between the analytical approach and the OpenFOAM simulation results for $\tilde{d} = 150 \mu\text{m}$ is approximately 0.38, for $\tilde{d} = 1550 \mu\text{m}$ it is approximately 0.06. Furthermore, the tail of the particle fraction after the peak drops sharply for the simulation results for $\tilde{d} = 150 \mu\text{m}$, while the tail tapers off slowly for the analytical approach. However, this effect evens out as the transmitter-to-mid-receiver distance \tilde{d} increases. Moreover, comparing Fig. 4a, Fig. 4b, and Fig. 4c, indicates that the larger the geometry’s extension in the fluid’s flow direction, the better the approximation of the OpenFOAM simulation results to the analytical approach, see also Section VII-B.

A comparison of the PBS results to the OpenFOAM simulation results indicates that for the case $a = 200 \mu\text{m}$ & $\tilde{d} = 150 \mu\text{m}$, the magnitude of the peak of the OpenFOAM simulation results is higher than the magnitude of the PBS results, i.e., $\text{CIR}_{\text{peak,avg,OpenFOAM}} / \text{CIR}_{\text{peak,avg,PBS}} = 1.26$. Comparing the peaks for the two cases $a = 200 \mu\text{m}$ & $\tilde{d} = 750 \mu\text{m}$ and $a = 200 \mu\text{m}$ & $\tilde{d} = 1550 \mu\text{m}$, we observe that the numerical difference in the fraction of the particles changes for the PBS compared to OpenFOAM; however, the shapes of the graphs are similar, i.e., the positions of the peaks and the decreasing tails are consistent between PBS and OpenFOAM. The $\text{CIR}_{\text{peak,avg,OpenFOAM}} / \text{CIR}_{\text{peak,avg,PBS}}$ ratios are 0.80 and 0.49 for cases (b) and (c) following Fig. 4, respectively.

B. ERROR CALCULATION

To evaluate the accuracy of the OpenFOAM microfluidic MC simulations, we calculate the RMSE for the OpenFOAM simulation results with respect to the analytical approach as well as for the OpenFOAM simulation results with respect to the PBS simulation results. To ensure the comparability of the RMSE values, the calculation of the RMSE values was started with the arrival of the first particle in the OpenFOAM simulation and carried out for a time interval Δt_{RMSE} of 1.2 s (corresponds to 60 written time steps of 0.02 s each).

For each evaluation scenario, we determine the RMSEs between the OpenFOAM simulation and the respective benchmarks (analytical approach, PBS). We determine the RMSEs as averages across ten simulation runs for the various transmitter-to-mid-receiver distances, i.e., $\tilde{d} = 150 \mu\text{m}$, $\tilde{d} = 750 \mu\text{m}$, and $\tilde{d} = 1550 \mu\text{m}$, as visualized in Fig. 5. Specifically, for a given simulation run, we determine the RMSE by comparing the fraction of the received particles for each time step. For the three cases, $\tilde{d} = 150 \mu\text{m}$,

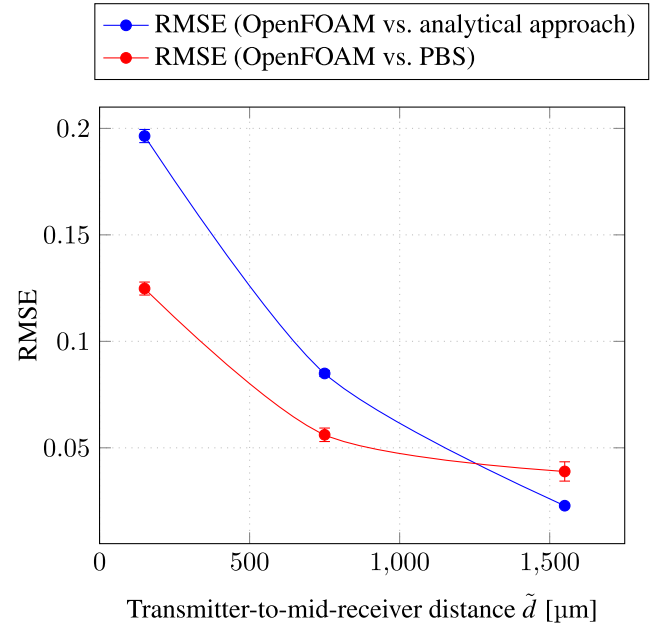


FIGURE 5. Root-Mean-Square Error (RMSE) for the three different transmitter-to-mid-receiver distances \tilde{d} . The connection between the points was approximated as a smooth graph. The 95 % confidence intervals are marked.

$\tilde{d} = 750 \mu\text{m}$, and $\tilde{d} = 1550 \mu\text{m}$, the RMSE (OpenFOAM vs. analytical approach) values are 19.64 %, 8.49 %, and 2.28 % of the absolute fraction of received particles in a time slot, respectively. Fig. 5 also indicates that as the transmitter-to-mid-receiver distance \tilde{d} increases, the simulation results converge non-linearly to the analytical approach.

Comparing the PBS results with the OpenFOAM-based simulation results in Fig. 5, the RMSE also decreases asymptotically with increasing transmitter-to-mid-receiver distance \tilde{d} . For the three cases, $\tilde{d} = 150 \mu\text{m}$, $\tilde{d} = 750 \mu\text{m}$, and $\tilde{d} = 1550 \mu\text{m}$, the RMSE (OpenFOAM vs. PBS) values are 12.48 %, 5.61 %, and 3.89 % of the absolute fraction of received particles in a time slot, respectively. Consequently, the RMSE values for $\tilde{d} = 150 \mu\text{m}$ and $\tilde{d} = 750 \mu\text{m}$ are higher for the analytical approach vs. the OpenFOAM-based simulation results than for OpenFOAM-based simulation results vs. the PBS results, cf. Fig. 5. This behavior changes after approximately $\tilde{d} = 1200 \mu\text{m}$ in Fig. 5. With an increasing \tilde{d} , the OpenFOAM simulation results for the peak’s magnitude value closely approximate the analytical approach, whereas the PBS gives significantly different peak magnitude values, cf. Fig. 4. Also, with an increasing \tilde{d} , the OpenFOAM simulation results closely approximate both the PBS and the analytical approach in the magnitude of the tail values, cf. Fig. 4.

Comparison of Fig. 4a, Fig. 4b, and Fig. 4c indicates that the peaks of the PBS and the OpenFOAM simulation results do not overlap. Aligning the peaks in PBSs and OpenFOAM simulations for MC may be necessary depending on the specific goals of future research. If the PBSs and OpenFOAM simulations are based on the same

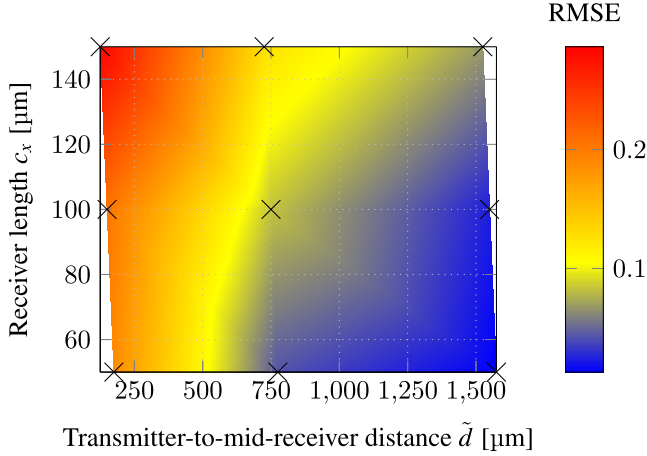


FIGURE 6. Influence of the transmitter-to-mid-receiver distance \tilde{d} and the length of the cylindrical receiver c_x on the Root-Mean-Square Error (RMSE) values for the OpenFOAM vs. the analysis RMSE values. Three different receiver lengths, $c_x = \{a/4, a/2, 3a/4\} = \{50, 100, 150\} \mu\text{m}$, were taken into account, marked with crosses.

physical principles and assumptions, it is sensible to align the peaks to ensure consistency so as to allow direct comparisons and interpretations of the results. Furthermore, aligning the peaks during data post-processing makes it easier to visualize and interpret the results, providing a clearer understanding of the data. The need for an alignment of the peaks could also depend on the specific characteristics of the MC channel. If the MC channel has well-defined peaks or features that are essential for channel analysis, aligning them might be necessary for accurate interpretation. Finally, any established future standards for aligning peaks in MC simulations within the scientific community should be considered (currently, there are no uniform standards, e.g., for coding for MC networks [10]). Adhering to community standards can enhance the reproducibility and comparability of simulation studies.

Fig. 6 shows the influences of the transmitter-to-mid-receiver distance \tilde{d} and the length c_x of the cylindrical receiver on the OpenFOAM vs. analysis RMSE values (evaluated for the fraction of received particles). The evaluation considers three different receiver lengths, $c_x = \{a/4, a/2, 3a/4\} = \{50, 100, 150\} \mu\text{m}$ and the corresponding transmitter-to-mid-receiver distances $\tilde{d}_{a/4} = \{175, 775, 1575\} \mu\text{m}$, $\tilde{d}_{a/2} = \{150, 750, 1550\} \mu\text{m}$, and $\tilde{d}_{3a/4} = \{125, 725, 1525\} \mu\text{m}$. The RMSE data points were visualized in Fig. 6 and interpolated between the data points for better visualization. Fig. 6 demonstrates that the greater the transmitter-to-mid-receiver distance \tilde{d} and the smaller the receiver length c_x , the smaller the RMSE values for OpenFOAM vs. analysis, colored in blue color.

C. TRADE-OFF BETWEEN EXECUTION TIME AND RMSE

We examine the trade-off between the execution time of the simulation runs and the RMSE by running OpenFOAM locally on a single core, i.e., not parallelized, and without

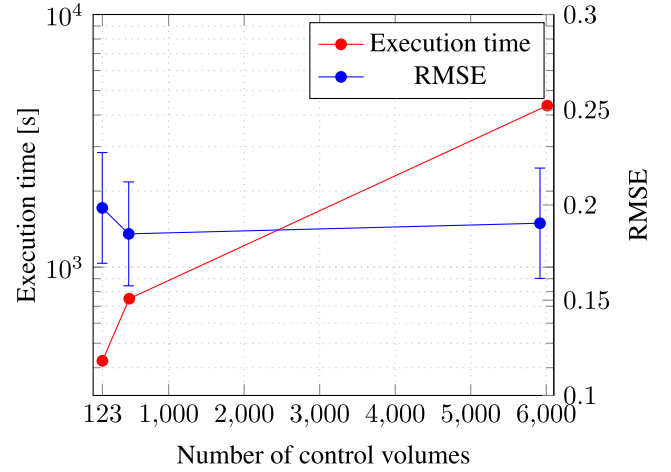


FIGURE 7. OpenFOAM execution time for the computation on one core and Root-Mean-Square Error (RMSE) of OpenFOAM vs. analysis with 95% confidence intervals for the ($a = 200 \mu\text{m}$ & $\tilde{d} = 150 \mu\text{m}$) case as a function of the mesh coarseness, i.e., the number of control volumes.

the dynamic adjustment of the time step (without the enabled parameter “AdjustTimeStep”), for the $d = 200 \mu\text{m}$ case. We created three meshes with different resolutions for the simulation case, resulting in three models with a total number of 123, 479, and 6014 control volumes, respectively. All other parameters were unchanged. We selected $1 \times 10^{-5} \text{ s}$ as the static time step size. Fig. 7 shows the execution time for the computation on one core and the RMSE compared to the analytical approach for a varying number of control volumes. With an increasing number of control volumes and, thus, a more finely resolved mesh, the execution time increases nearly linearly. Concomitantly, Fig. 7 indicates that the RMSE values increase very slightly with an increasing number of control volumes after a slight drop in the medium mesh resolution model. However, the RMSE values (OpenFOAM vs. analysis) are approximately constant, obtaining an acceptable result even with fewer control volumes.

We note that PBS does not have a mesh, i.e., does not have different numbers of control volumes. We observed that for the $d = 200$ case, PBS has an execution time below 1 second, i.e., is orders of magnitude faster than OpenFOAM. However, PBS has several drawbacks compared to OpenFOAM. PBS is typically less accurate than methods based on solving partial derivative equations for control volumes, such as in OpenFOAM, particularly for capturing small-scale turbulence [79]. Also, complex boundary conditions, turbulent flows, and complex geometries are typically more challenging in PBS than in OpenFOAM [79].

VIII. EXPERIMENTAL TESTBED RESULTS

To verify our simulation results, i.e., to compare the OpenFOAM simulation results to experimental testbed measurements, we use the experimental setup from [18] in the millimeter scale; see Fig. 8. Specifically, we focus on the flow-dominated regime, which governs the testbed setup as per Eq. (11), $v_{\text{eff}} r_{\text{ch}} / D = 18881.76 \gg 208.96 = 4 d_{\text{ch}} / r_{\text{ch}}$.

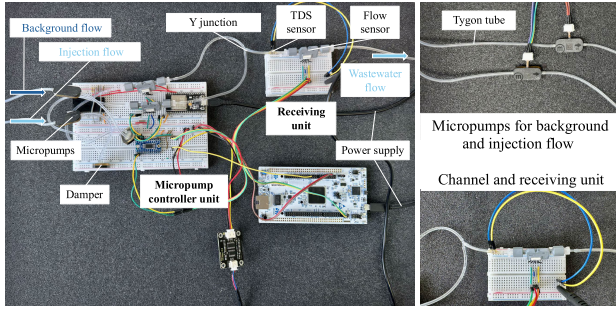


FIGURE 8. Experimental testbed setup [80]. The experimental testbed setup is a slightly modified version from [18].

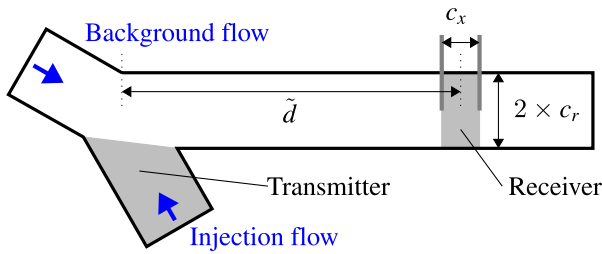


FIGURE 9. Channel characteristics of the experimental testbed setup, serving as a basis for the meshing in OpenFOAM; adapted from [80].

A. EXPERIMENTAL TESTBED SETUP

We utilize the experimental testbed setup from Angerbauer et al. [18] to obtain the experimental measurement data. The experimental setup comprises three main components: the transmitter, the channel, and the receiver, cf. Figs. 8 and 9. The system is designed to modulate messages into m varying concentration levels of a salt solution, specifically a sodium chloride solution. Specifically, we inject a salt solution with a concentration of 9 g L^{-1} at the transmitter side into a background flow of distilled water. Through an active drift mechanism involving a background flow of distilled water, the injected ions are transported to the receiver at the channel's outlet. The receiver utilizes a conductivity sensor to demodulate the messages, whereby higher salt concentrations result in increased conductivity readings. All technical specifications can be found in [80].

Fig. 9 shows the channel characteristics for the experimental setup, which also serve as the basis for the meshing in the OpenFOAM simulation environment. The channel physically consists of a Y-junction and a Tygon tube. As recommended by Bartunik et al. [81], we arrange the direction of the injection flow to oppose the background flow, thereby preventing particles from being swept away from the injection surface. On the receiver end, two miniature electrodes are embedded within the channel, and the separation between them is labeled as c_x . The parameter \tilde{d} represents the channel length, i.e., the distance from the transmitter (TX) to mid-receiver (RX), and c_r represents the radius of the channel.

To obtain the experimental data, we conducted $n_{\text{exp}} = 10$ measurement replication runs. The salt solution was injected

TABLE 5. Parameters of the experimental setup; adapted from [80].

Parameter	Symbol	Value	Unit
Salt concentration	c_{NaCl}	9.0	g/L
Radius of the channel	c_r	0.67	mm
Distance from TX to RX	\tilde{d}	35	mm
Distance of RX electrodes	c_x	1	mm
Background flow rate	$Q_{\text{background}}$	3.1	mL/min
Injection flow rate	Q_{inject}	3.1	mL/min
Injection pulse duration	T_{inject}	0.05	s
Number of runs	n_{exp}	10	—

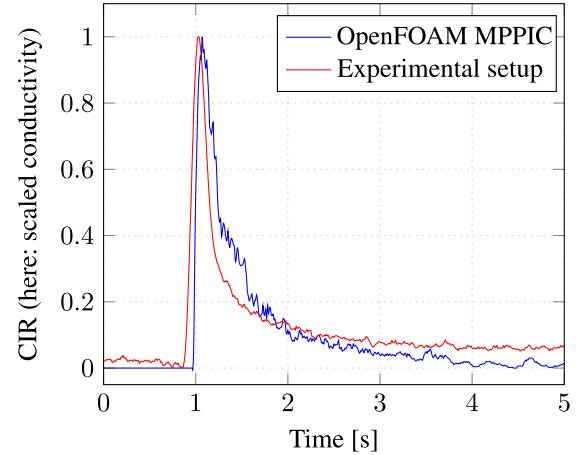


FIGURE 10. Comparison of the OpenFOAM MPPIC solver simulation results versus the measurement data obtained from the experimental testbed setup for a background flow rate of 3.1 mL/min, i.e., flow-dominated regime.

once in a run into the background flow with an injection pulse duration of $T_{\text{inject}} = 0.05 \text{ s}$. The experimentally obtained CIR can be seen in Fig. 10.

B. EXPERIMENTAL VERSUS SIMULATION RESULTS

Fig. 10 compares the experimentally obtained and simulated data using OpenFOAM. We only consider the flow-dominated regime. Clearly, the two CIRs are close to each other, and the peaks match well. There are slight differences in the course of the tail. The experimentally obtained data exhibit a higher number of received particles, especially in the tail's run-out region, compared to the simulation results. A main reason for this is the lack of support for the MPPIC solver for diffusion, which we will consider and implement in future research. The difference can be quantified with an RMSE of 9.89 %, which signifies that the average difference between the experimentally obtained data and the simulation data is below 10 % of the data range. The RMSE value here was obtained in the full data range, i.e., 0 s to 5 s, cf. Fig. 10, as we already observed arriving particles before the peak arises. To summarize, the testbed measurements validate that the MPPIC solver can simulate flow-dominated regimes in microfluidic MC.

IX. DISCUSSION AND LIMITATIONS

This section discusses the results, points out open problems, and highlights limitations, especially of the MPPIC solver.

A. DISCUSSION AND OPEN PROBLEMS

The analytical approach, proposed by Wicke et al. [16] in 2018, serves in our study as an approximation of the true physical reality. However, the analytical model has not been directly compared to the physical reality in [16], thus, the extent of the error inherent in this approximation of the physical reality remains unknown. Consequently, it becomes challenging to compute an accurate assessment of the cumulative error introduced by the simulation model employed in the microfluidic MC context. Nevertheless, the analytical model [16] has been commonly used for characterizing microfluidic MC [23], [82], [83]. For instance, Schafer et al. [42] compare their proposed models for transfer functions for cylindrical MC channels with diffusion and laminar flow with the flow dominant regimes from [16]. Also, Jing et al. [84] investigate a mobile MC system under consideration of the dispersion regime from [16].

The lack of empirical data collected from experimental setup measurements compounds the issue. These measurements are crucial for comprehensively evaluating the simulation and analytical models. Without such an empirical validation, the validity and reliability of the models in representing the actual physical system cannot be substantiated. Facing this issue, we compared the MPPIC solver to an experimental testbed setup in the millimeter scale. The measurements on the experimental setup serve as an initial guide but must be compared with measurements on the microscopic scale in future studies due to different flow dynamics in both regimes [85].

We considered an elementary tube geometry and water. Future simulations for the IoBNT should examine more complex geometries, e.g., real-world cardiovascular models, e.g., from [86], blood as a fluid [70], as well as boundary conditions at the inlet for time-dependent volumetric flows. The presented study serves as a first feasibility and validation study that such future studies can build on.

In the present study, we have focused on the flow-dominated system, with no consideration given to the dispersion system, see Section IV. Two key points emerge from this approach: firstly, future research should address the dispersion system, similar to the way that we have studied the flow-dominated system; and, secondly, it is worth noting that even in a flow-dominated system, small dispersion exists (which we neglected).

A comparison of the computation times for the OpenFOAM-based simulation with the MPPIC solver and the PBS also indicates significant differences. While the OpenFOAM computation times (on the HPC) are several hours for $d = 200 \mu\text{m}$, $d = 800 \mu\text{m}$, and $d = 1600 \mu\text{m}$, the PBS computation times, based on *Matlab*, are below 1 second. However, as observed from Figure 7, the computation time of the OpenFOAM simulation depends strongly on the mesh resolution. The meshes in our study are very highly resolved, resulting in correspondingly high computing times. The advantages of the OpenFOAM-based MPPIC solver approach with respect to the PBS arise especially in the simulation of

more complex geometries (e.g., of human arteries [87]), other fluid or particle parameters, but also for other inlet and outlet boundary conditions.

Future research should examine the extent to which simulation models can be verified using measurements on experimental setups, e.g., in lab-on-a-chip systems or microfluidic testbeds [18]. In particular, for medical applications, it is important to verify whether the simulations achieve sufficiently accurate results, e.g., for targeted drug transport in the human body. In this study, we only investigated laminar flow; future research should aim at turbulent flows and, thus, the turbulence modeling, including the analytical description of the turbulence. In addition, analytical models should be adjusted as necessary as simulation results and experimental setup verifications progress.

B. SOLVER LIMITATIONS

The MPPIC solver assumes that the particles are spherical. Incorporating adjustments for non-spherical particles is feasible by adapting the particle drag model; however, when dealing with extremely non-spherical particles, the model may not accurately depict the actual interactions [88]. Furthermore, if the particle size is on a similar scale as the size of the control volume, or when the precise capture of particle collisions is compromised, a more refined mesh is necessary to accurately simulate the fluid flow [88].

C. OpenFOAM AS A CFD TOOL FOR MICROFLUIDIC MC

OpenFOAM offers significant advantages for CFD simulations due to its open-source nature and extensive capabilities. Compared to commercial CFD software tools, such as COMSOL or ANSYS, the freely accessible OpenFOAM source code allows for customization and modification of solvers, enabling researchers to tailor simulations to their specific needs [89]. The OpenFOAM pre- and post-processing interfaces are robust and user-friendly [90]. Also, OpenFOAM supports dynamic meshes, allowing the handling and manipulation of moving grids in simulations, i.e., involving dynamically changing geometries [90]. Another advantage of OpenFOAM is the very good parallelizability of computations, which makes it possible to perform computationally intensive calculations, as conducted in this study. Furthermore, OpenFOAM utilizes a user-friendly syntax for defining partial differential equations, streamlining the development of new models for complex physical phenomena [89].

X. CONCLUSION AND FUTURE RESEARCH

We have developed a novel methodology to simulate microfluidic Molecular Communication (MC) in the flow-dominated regime using the OpenFOAM Multi-Phase Particle-In-Cell (MPPIC) solver and have made the corresponding scripts publicly available [19]. Initially designed for sedimentation, separation, fluidized beds (e.g., combustion and catalytic cracking), and other granular

flows, we examined the MPPIC solver's applicability to microfluidic flow in this methods article. We compared the OpenFOAM simulation results, specifically the Channel Impulse Response (CIR), with an analytical flow-dominated approach, with Particle-Based Simulation (PBS) results, and with measurements from an experimental testbed. Comparing the OpenFOAM simulations with the analytical approach revealed similar shapes and maximum particle fractions. However, disparities were observed in the onset and decay of the peak. Additionally, a comparison with PBS results showed close agreement for smaller transmitter-receiver distances (e.g., 200 μm), with discrepancies increasing with greater distances, at times up to a factor of roughly two.

However, the greater the distance between the transmitter and the receiver, the smaller the gap in the tail of the CIRs between the OpenFOAM simulation results, the PBS, and the analytical approach. The calculation of the Root-Mean-Square Error (RMSE) values indicated a non-linear convergence of the OpenFOAM results to both the analytical approach and the PBS with increasing distance, while the RMSE values for the comparison of the OpenFOAM simulation results vs. the analytical approach are lower than the OpenFOAM vs. PBS RMSE values. Comparing the introduced MPPIC solver in the millimeter scale to measurements on an experimental testbed setup, demonstrated a convergence behavior for the flow-dominated regime. Future research includes comparing the simulation results to experimental setup measurements in the microscale domain.

Future investigations should explore more intricate geometries, such as models resembling the human bloodstream, turbulent flows induced by obstacles (e.g., plaques in blood vessels [70]), and tailored fluid properties mimicking human blood [87]. Unlike the *Newtonian* fluid water used in our simulations, blood exhibits non-*Newtonian* behavior. Moreover, adjusting inlet and outlet conditions to replicate the cardiac cycle of humans is essential [87]. Further research avenues include examining the diffusion-dominated scenario (which includes adapting the solver to the characteristics of the diffusion-based regime for smoother flow), omitted in this preliminary study, and validating simulation results against real-world experimental data. Future simulations may also explore alternative modulation or coding techniques [10], [91], [92], or even simulate entire bit sequences to broaden the scope of our understanding. Furthermore, as the role of Computational Fluid Dynamics (CFD) simulations in microfluidic MC is still largely unexplored, future studies should comprehensively compare the existing CFD simulation tools, such as ANSYS Fluent, COMSOL Multiphysics, and OpenFOAM, for a wide range of geometries, setups, and channels.

ACKNOWLEDGMENT

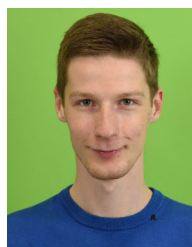
The authors gratefully acknowledge the GWK support through computing time at the Center for Information Services and HPC (ZIH) at TU Dresden.

REFERENCES

- [1] I. F. Akyildiz, M. Pierobon, S. Balasubramaniam, and Y. Koucheryavy, "The Internet of Bio-Nano Things," *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 32–40, Mar. 2015.
- [2] S. Zafar, M. Nazir, T. Bakhshi, H. A. Khattak, S. Khan, M. Bilal, K. R. Choo, K.-S. Kwak, and A. Sabah, "A systematic review of bio-cyber interface technologies and security issues for Internet of Bio-Nano Things," *IEEE Access*, vol. 9, pp. 93529–93566, 2021.
- [3] C. Lee, B.-H. Koo, C.-B. Chae, and R. Schober, "The Internet of Bio-Nano Things in blood vessels: System design and prototypes," *J. Commun. Netw.*, vol. 25, no. 2, pp. 222–231, Apr. 2023.
- [4] A. Etemadi, M. Farahnak-Ghazani, H. Arjmandi, M. Mirmohseni, and M. Nasiri-Kenari, "Abnormality detection and localization schemes using molecular communication systems: A survey," *IEEE Access*, vol. 11, pp. 1761–1792, 2023.
- [5] C. D. M. Regis, I. D. S. Silva, P. I. A. Guimarães, E. T. D. Silva, and M. T. Barros, "Dual ionic transport using Ca^{2+} and Na^{2+} signaling for molecular communication systems," *IEEE Access*, vol. 11, pp. 61331–61345, 2023.
- [6] K. Yang, D. Bi, Y. Deng, R. Zhang, M. M. U. Rahman, N. A. Ali, M. A. Imran, J. M. Jornet, Q. H. Abbasi, and A. Alomainy, "A comprehensive survey on hybrid communication in context of molecular communication and terahertz communication for body-centric nanonetworks," *IEEE Trans. Mol., Biol. Multi-Scale Commun.*, vol. 6, no. 2, pp. 107–133, Nov. 2020.
- [7] C. Koca and O. B. Akan, "Channel clearance by perfectly absorbing boundaries in synaptic molecular communications," *IEEE Access*, vol. 10, pp. 121480–121493, 2022.
- [8] A. O. Bicen and I. F. Akyildiz, "System-theoretic analysis and least-squares design of microfluidic channels for flow-induced molecular communication," *IEEE Trans. Signal Process.*, vol. 61, no. 20, pp. 5000–5013, Oct. 2013.
- [9] H. Abin, A. Gohari, and M. Nasiri-Kenari, "A study of chemical reactions in point-to-point diffusion-based molecular communication," *IEEE Access*, vol. 11, pp. 24752–24767, 2023.
- [10] P. Hofmann, J. A. Cabrera, R. Bassoli, M. Reisslein, and F. H. P. Fitzek, "Coding in diffusion-based molecular nanonetworks: A comprehensive survey," *IEEE Access*, vol. 11, pp. 16411–16465, 2023.
- [11] W. Wicke, A. Ahmadzadeh, V. Jamali, H. Unterwieser, C. Alexiou, and R. Schober, "Magnetic nanoparticle-based molecular communication in microfluidic environments," *IEEE Trans. Nanobiosci.*, vol. 18, no. 2, pp. 156–169, Apr. 2019.
- [12] N. Farsad, A. W. Eckford, S. Hiyama, and Y. Moritani, "On-chip molecular communication: Analysis and design," *IEEE Trans. Nanobiosci.*, vol. 11, no. 3, pp. 304–314, Sep. 2012.
- [13] M. Bartunik, M. Fleischer, W. Haselmayr, and J. Kirchner, "Colour-specific microfluidic droplet detection for molecular communication," in *Proc. 7th ACM Int. Conf. Nanosc. Comput. Commun.*, Sep. 2020, pp. 1–6.
- [14] N. Farsad, H. B. Yilmaz, A. Eckford, C.-B. Chae, and W. Guo, "A comprehensive survey of recent advancements in molecular communication," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1887–1919, 3rd Quart., 2016.
- [15] D. Bi, Y. Deng, M. Pierobon, and A. Nallanathan, "Chemical reactions-based microfluidic transmitter and receiver design for molecular communication," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5590–5605, Sep. 2020.
- [16] W. Wicke, T. Schwering, A. Ahmadzadeh, V. Jamali, A. Noel, and R. Schober, "Modeling duct flow for molecular communication," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 206–212.
- [17] M. Andrews and P. O'Rourke, "The multiphase particle-in-cell (MP-PIC) method for dense particulate flows," *Int. J. Multiphase Flow*, vol. 22, no. 2, pp. 379–402, Apr. 1996.
- [18] S. Angerbauer, M. Hamidovic, F. Enzenhofer, M. Bartunik, J. Kirchner, A. Springer, and W. Haselmayr, "Salinity-based molecular communication in microfluidic channels," *IEEE Trans. Mol., Biol. Multi-Scale Commun.*, vol. 9, no. 2, pp. 191–206, Jun. 2023.
- [19] P. Hofmann, P. Zhou, C. Lee, M. Reisslein, F. H. P. Fitzek, and C.-B. Chae, (Oct. 2023). *Dataset for the Simulation of Microfluidic Molecular Communication Using OpenFOAM*. [Online]. Available: <https://iee-dataport.org/documents/dataset-simulation-microfluidic-molecular-communication-using-openfoam>
- [20] A. O. Bicen and I. F. Akyildiz, "End-to-end propagation noise and memory analysis for molecular communication over microfluidic channels," *IEEE Trans. Commun.*, vol. 62, no. 7, pp. 2432–2443, Jul. 2014.

- [21] A. O. Bicen and I. F. Akyildiz, "Interference modeling and capacity analysis for microfluidic molecular communication channels," *IEEE Trans. Nanotechnol.*, vol. 14, no. 3, pp. 570–579, May 2015.
- [22] F. Dinc, B. C. Akdeniz, A. E. Pusane, and T. Tugcu, "A general analytical approximation to impulse response of 3-D microfluidic channels in molecular communication," *IEEE Trans. Nanobiosci.*, vol. 18, no. 3, pp. 396–403, Jul. 2019.
- [23] V. Jamali, A. Ahmadzadeh, W. Wicke, A. Noel, and R. Schober, "Channel modeling for diffusive molecular communication—A tutorial review," *Proc. IEEE*, vol. 107, no. 7, pp. 1256–1301, Jul. 2019.
- [24] M. Hamidović, S. Angerbauer, D. Bi, Y. Deng, T. Tugcu, and W. Haselmayr, "Microfluidic systems for molecular communications: A review from theory to practice," *IEEE Trans. Mol., Biol., Multi-Scale Commun.*, vol. 10, no. 1, pp. 147–163, Mar. 2024.
- [25] Y. Jian, B. Krishnaswamy, C. M. Austin, A. O. Bicen, A. Einolghozati, J. E. Perdomo, S. C. Patel, F. Fekri, I. F. Akyildiz, C. R. Forest, and R. Sivakumar, "nanoNS3: A network simulator for bacterial nanonetworks based on molecular communication," *Nano Commun. Netw.*, vol. 12, pp. 1–11, Jun. 2017.
- [26] B. Drawert, S. Engblom, and A. Hellander, "URDME: A modular framework for stochastic simulation of reaction-transport processes in complex geometries," *BMC Syst. Biol.*, vol. 6, no. 1, p. 76, Jun. 2012.
- [27] M. Ander, I. Tomás-Oliveira, J. Ferkinghoff-Borg, P. Beltrao, M. Foglierini, B. Di Ventura, L. Serrano, and C. Lemerle, "SmartCell, a framework to simulate cellular processes that combines stochastic approximation with diffusion and localisation: Analysis of simple networks," *Syst. Biol.*, vol. 1, no. 1, pp. 129–138, Jun. 2004.
- [28] J. Hattné, D. Fange, and J. Elf, "Stochastic reaction-diffusion simulation with MesoRD," *Bioinformatics*, vol. 21, no. 12, pp. 2923–2924, Apr. 2005.
- [29] G. Wei, P. Bogdan, and R. Marculescu, "Efficient modeling and simulation of bacteria-based nanonetworks with BNSim," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, pp. 868–878, Dec. 2013.
- [30] N. A. Turgut, B. A. Bilgin, and O. B. Akan, "N⁴Sim: The first nervous NaNoNetwork simulator with synaptic molecular communications," *IEEE Trans. Nanobiosci.*, vol. 21, no. 4, pp. 468–481, Oct. 2022.
- [31] S. S. Andrews, N. J. Addy, R. Brent, and A. P. Arkin, "Detailed simulations of cell biology with Smoldyn 2.1," *PLoS Comput. Biol.*, vol. 6, no. 3, Mar. 2010, Art. no. e1000705.
- [32] L. Felicetti, M. Femminella, and G. Reali, "Simulation of molecular signaling in blood vessels: Software design and application to atherogenesis," *Nano Commun. Netw.*, vol. 4, no. 3, pp. 98–119, Sep. 2013.
- [33] I. Llatser, I. Pascual, N. Garralda, A. Cabellos-Aparicio, M. Pierobon, E. Alarcon, and J. Sole-Pareta, "Exploring the physical channel of diffusion-based molecular communication by simulation," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2011, pp. 1–5.
- [34] I. Llatser, D. Demiray, A. Cabellos-Aparicio, D. T. Altılar, and E. Alarcón, "N3Sim: Simulation framework for diffusion-based molecular communication nanonetworks," *Simul. Model. Pract. Theory*, vol. 42, pp. 210–222, Mar. 2014.
- [35] A. Noel, K. C. Cheung, R. Schober, D. Makrakis, and A. Hafid, "Simulating with AcCoRD: Actor-based communication via reaction-diffusion," *Nano Commun. Netw.*, vol. 11, pp. 44–75, Mar. 2017.
- [36] L. A. Harris, J. S. Hogg, J.-J. Tapia, J. A. P. Sekar, S. Gupta, I. Korsunsky, A. Arora, D. Barua, R. P. Sheehan, and J. R. Faeder, "BioNetGen 2.2: Advances in rule-based modeling," *Bioinformatics*, vol. 32, no. 21, pp. 3366–3368, Jul. 2016.
- [37] Y. Wang, A. Noel, and N. Yang, "A novel a priori simulation algorithm for absorbing receivers in diffusion-based molecular communication systems," *IEEE Trans. Nanobiosci.*, vol. 18, no. 3, pp. 437–447, Jul. 2019.
- [38] H. B. Yilmaz and C.-B. Chae, "Simulation study of molecular communication systems with an absorbing receiver: Modulation and ISI mitigation techniques," *Simul. Model. Pract. Theory*, vol. 49, pp. 136–150, Dec. 2014.
- [39] H. Jasak, "OpenFOAM: Open source CFD in research and industry," *Int. J. Nav. Archit. Ocean Eng.*, vol. 1, no. 2, pp. 89–94, Dec. 2009.
- [40] COMSOL Multiphysics. *Simulate Real-World Designs, Devices, and Processes With Multiphysics Software From COMSOL*. Accessed: May 26, 2024. [Online]. Available: <https://www.comsol.com/>
- [41] ANSYS. *Ansys Fluent: Fluid Simulation Software*. Accessed: May 26, 2024. [Online]. Available: <https://www.ansys.com/de/products/fluids/ansys-fluent>
- [42] M. Schäfer, W. Wicke, L. Brand, R. Rabenstein, and R. Schober, "Transfer function models for cylindrical MC channels with diffusion and laminar flow," *IEEE Trans. Mol., Biol. Multi-Scale Commun.*, vol. 7, no. 4, pp. 271–287, Dec. 2021.
- [43] M. Kuscü and O. B. Akan, "Modeling convection-diffusion-reaction systems for microfluidic molecular communications with surface-based receivers in Internet of Bio-Nano Things," *PLoS ONE*, vol. 13, no. 2, Feb. 2018, Art. no. e0192202.
- [44] P. Gharpure, S. Veeralingam, and S. Badhulika, "Bio-inspired uniform flow microfluidic sensor platform for multi-analyte sensing: A simulation-based outflow and injection study," *Microfluidics Nanofluidics*, vol. 25, no. 10, Sep. 2021.
- [45] A. Zeshan, A. Emre Pusane, and T. Baykas, "An integrated molecular communication system based on acoustic tweezers," *IEEE Trans. Nanobiosci.*, vol. 22, no. 2, pp. 413–419, Apr. 2023.
- [46] G. Fink, F. Costamoling, and R. Wille, "MMFT droplet simulator: Efficient simulation of droplet-based microfluidic devices," *Softw. Impacts*, vol. 14, Dec. 2022, Art. no. 100440.
- [47] J. P. Drees, L. Stratmann, F. Bronner, M. Bartunik, J. Kirchner, H. Unterwiesing, and F. Dressler, "Efficient simulation of macroscopic molecular communication: The pogo simulator," in *Proc. 7th ACM Int. Conf. Nanosc. Comput. Commun.*, Sep. 2020, pp. 1–6.
- [48] R. I. Issa, "Solution of the implicitly discretised fluid flow equations by operator-splitting," *J. Comput. Phys.*, vol. 62, no. 1, pp. 40–65, Jan. 1986.
- [49] Ž. Tuković, A. Karač, P. Cardiff, H. Jasak, and A. Ivanković, "OpenFOAM finite volume solver for fluid-solid interaction," *Trans. FAMENA*, vol. 42, no. 3, pp. 1–31, Oct. 2018.
- [50] M. Kuscü, H. Ramezani, E. Dinc, S. Akhavan, and O. B. Akan, "Fabrication and microfluidic analysis of graphene-based molecular communication receiver for Internet of Nano Things (IoNT)," *Sci. Rep.*, vol. 11, no. 1, p. 19600, Oct. 2021.
- [51] F. Cali, L. Fichera, G. Trusso Sfrassetto, G. Nicotra, G. Sfuncia, E. Bruno, L. Lanzano, I. Barbagallo, G. Li-Destri, and N. Tuccitto, "Fluorescent nanoparticles for reliable communication among implantable medical devices," *Carbon*, vol. 190, pp. 262–275, Apr. 2022.
- [52] F. Cali, G. Li-Destri, and N. Tuccitto, "Interfacial shift keying allows a high information rate in molecular communication: Methods and data," *IEEE Trans. Mol., Biol. Multi-Scale Commun.*, vol. 9, no. 3, pp. 300–307, Sep. 2023.
- [53] V. Walter, D. Bi, A. Salehi-Reyhani, and Y. Deng, "Real-time signal processing via chemical reactions for a microfluidic molecular communication system," *Nature Commun.*, vol. 14, no. 1, p. 7188, Nov. 2023.
- [54] M. Bartunik, G. Fischer, and J. Kirchner, "The development of a biocompatible testbed for molecular communication with magnetic nanoparticles," *IEEE Trans. Mol., Biol. Multi-Scale Commun.*, vol. 9, no. 2, pp. 179–190, Jun. 2023.
- [55] M. Girfoglio, A. Quaini, and G. Rozza, "A linear filter regularization for POD-based reduced-order models of the quasi-geostrophic equations," *Comp. Rendus. Mécanique*, vol. 351, no. 1, pp. 457–477, Apr. 2024.
- [56] E. S. Menon, "Chapter five—Fluid flow in pipes," in *Transmission Pipeline Calculations and Simulations Manual*, E. S. Menon, Ed., Boston, MA, USA: Gulf Professional, 2015, pp. 149–234.
- [57] H. Bruus, *Theoretical Microfluidics*, vol. 18. Oxford, U.K.: Oxford Univ. Press, Jan. 2007.
- [58] H. B. Yilmaz, A. C. Heren, T. Tugcu, and C.-B. Chae, "Three-dimensional channel characteristics for molecular communications with an absorbing receiver," *IEEE Commun. Lett.*, vol. 18, no. 6, pp. 929–932, Jun. 2014.
- [59] R. F. Probstein, *Physicochemical Hydrodynamics*. Hoboken, NJ, USA: Wiley, Feb. 2005.
- [60] I. Cho, C. Yang, H. Kwon, B. Bang, S. Jeong, and U. Lee, "Numerical simulation method of a circulating fluidized bed reactor using a modified MP-PIC solver of OpenFOAM," *Powder Technol.*, vol. 409, Sep. 2022, Art. no. 117815.
- [61] C. Greenshields. (2018). *OpenFOAM V6 User Guide*. CFD Direct. [Online]. Available: <https://doc.cfd.direct/openfoam/user-guide-v6/>
- [62] D. M. Snider, "An incompressible three-dimensional multiphase Particle-in-Cell model for dense particle flows," *J. Comput. Phys.*, vol. 170, no. 2, pp. 523–549, Jul. 2001.
- [63] D. M. Snider, S. M. Clark, and P. J. O'Rourke, "Eulerian-Lagrangian method for three-dimensional thermal reacting flow with application to coal gasifiers," *Chem. Eng. Sci.*, vol. 66, no. 6, pp. 1285–1295, Mar. 2011.

- [64] P. J. O'Rourke and D. M. Snider, "A new blended acceleration model for the particle contact forces induced by an interstitial fluid in dense particle/fluid flows," *Powder Technol.*, vol. 256, pp. 39–51, Apr. 2014.
- [65] S. E. Harris and D. G. Crighton, "Solitons, solitary waves, and voidage disturbances in gas-fluidized beds," *J. Fluid Mech.*, vol. 266, pp. 243–276, May 1994.
- [66] M. T. Tyn and T. W. Gusek, "Prediction of diffusion coefficients of proteins," *Biotechnol. Bioengineering*, vol. 35, no. 4, pp. 327–338, Feb. 1990.
- [67] S. Ergun, "Fluid flow through packed columns," *Chem. Eng. Prog.*, vol. 48, no. 2, pp. 88–95, Feb. 1952.
- [68] C. Y. Wen and Y. H. Yu, "Mechanics of fluidization," in *Proc. Chem. Eng. Prog. Symp. Ser.*, 1966, vol. 62, no. 2, pp. 100–111.
- [69] M. L. Huber, R. A. Perkins, A. Laescke, D. G. Friend, J. V. Sengers, M. J. Assael, I. N. Metaxa, E. Vogel, R. Mares, and K. Miyagawa, "New international formulation for the viscosity of H₂O," *J. Phys. Chem. Ref. Data*, vol. 38, no. 2, pp. 101–125, Apr. 2009.
- [70] P. Hofmann, S. Schmidt, A. Wietfeld, P. Zhou, J. Fuchtmann, F. H. P. Fitzek, and W. Kellerer, "A molecular communication perspective on detecting arterial plaque formation," *IEEE Trans. Mol., Biol. Multi-Scale Commun.*, early access, Jul. 4, 2024, doi: 10.1109/TMBMC.2024.3423005.
- [71] C. Greenshields and H. Weller, *Notes on Computational Fluid Dynamics: General Principles*. Reading, U.K.: CFD Direct, 2022.
- [72] J. Anderson, *Computational Fluid Dynamics: The Basics With Applications*. New York, NY, USA: McGraw-Hill, 1995.
- [73] N. W. Ryan and M. M. Johnson, "Transition from laminar to turbulent flow in pipes," *AIChE J.*, vol. 5, no. 4, pp. 433–435, Dec. 1959.
- [74] H. P. Erickson, "Size and shape of protein molecules at the nanometer level determined by sedimentation, gel filtration, and electron microscopy," *Biol. Procedures Online*, vol. 11, no. 1, pp. 32–51, May 2009.
- [75] E. Ponder, "The relation between red blood cell density and corpuscular hemoglobin concentration," *J. Biol. Chem.*, vol. 144, no. 2, pp. 333–338, Jul. 1942.
- [76] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, "A tensorial approach to computational continuum mechanics using object-oriented techniques," *Comput. Phys.*, vol. 12, no. 6, pp. 620–631, Nov. 1998.
- [77] S. Tanprasert, N. Rangton, W. Nukkhong, P. Wises, P. Piumsombon, and B. Chalermisinsuwan, "Impact of biomass fuel feeding ratio in co-firing circulating fluidized bed boiler: A computational fluid dynamics study," *J. Adv. Res. Numer. Heat Transf.*, vol. 17, no. 1, pp. 44–54, Mar. 2024.
- [78] R. Courant, K. Friedrichs, and H. Lewy, "Über die partiellen differenzengleichungen der mathematischen Physik," *Mathematische Annalen*, vol. 100, no. 1, pp. 32–74, Dec. 1928.
- [79] D. Violeau and B. D. Rogers, "Smoothed particle hydrodynamics (SPH) for free-surface flows: Past, present and future," *J. Hydraulic Res.*, vol. 54, no. 1, pp. 1–26, Jan. 2016.
- [80] P. Zhou, J. A. Cabrera, P. Hofmann, and F. H. P. Fitzek, "Detection algorithm study for a microfluidic molecular communication testbed setup," in *Proc. 8th Workshop Mol. Commun.*, Apr. 2024, pp. 1–5.
- [81] M. Bartunik, J. Teller, G. Fischer, and J. Kirchner, "Channel parameter studies of a molecular communication testbed with biocompatible information carriers: Methods and data," *IEEE Trans. Mol., Biol., Multi-Scale Commun.*, vol. 9, no. 4, pp. 489–498, Dec. 2023.
- [82] D. Bi, A. Almpanis, A. Noel, Y. Deng, and R. Schober, "A survey of molecular communication in cell biology: Establishing a new hierarchy for interdisciplinary applications," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1494–1545, 3rd Quart., 2021.
- [83] Y.-F. Lo, C.-H. Lee, P.-C. Chou, and P.-C. Yeh, "Modeling molecular communications in tubes with Poiseuille flow and Robin boundary condition," *IEEE Commun. Lett.*, vol. 23, no. 8, pp. 1314–1318, Aug. 2019.
- [84] D. Jing, Y. Li, and A. W. Eckford, "An extended Kalman filter for distance estimation and power control in mobile molecular communication," *IEEE Trans. Commun.*, vol. 70, no. 7, pp. 4373–4385, Jul. 2022.
- [85] Y. Song, D. Cheng, and L. Zhao, *Microfluidics: Fundamentals, Devices, and Applications*. Hoboken, NJ, USA: Wiley, Jan. 2018.
- [86] N. M. Wilson, A. K. Ortiz, and A. B. Johnson, "The vascular model repository: A public resource of medical imaging data and blood flow simulation results," *J. Med. Devices*, vol. 7, no. 4, Dec. 2013, Art. no. 040923.
- [87] S. Yogeswaran and F. Liu, "Vascular flow simulations using SimVascular and OpenFOAM," *MedRxiv*, pp. 1–15, Sep. 2021.
- [88] U. Caliskan and S. Miskovic, "A chimera approach for MP-PIC simulations of dense particulate flows using large parcel size relative to the computational cell size," *Chem. Eng. J. Adv.*, vol. 5, Mar. 2021, Art. no. 100054.
- [89] SimFlow Computational Fluid Dynamics (CFD) Software. *What is OpenFOAM*. Accessed: May 25, 2024. [Online]. Available: <https://simflow.com/what-is-openfoam/>
- [90] G. Chen, Q. Xiong, P. J. Morris, E. G. Paterson, A. Sergeev, and Y. Wang, "OpenFOAM for computational fluid dynamics," *Notices AMS*, vol. 61, no. 4, pp. 354–363, Apr. 2014.
- [91] N.-R. Kim and C.-B. Chae, "Novel modulation techniques using isomers as messenger molecules for nano communication networks via diffusion," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, pp. 847–856, Dec. 2013.
- [92] M. S. Kuran, H. B. Yilmaz, I. Demirkol, N. Farsad, and A. Goldsmith, "A survey on modulation techniques in molecular communication via diffusion," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 7–28, 1st Quart., 2021.



molecular communications, biological computing, and nanonetworking.



Dresden, Germany. His research interests include molecular communication, liquid-based experimental setups for molecular communications, and coding and modeling for diffusive molecular communications.



CHANGMIN LEE (Member, IEEE) received the B.S. and Ph.D. degrees from the School of Integrated Technology, Yonsei University, Seoul, South Korea, in 2015 and 2023, respectively. Since 2023, he has been a Senior Engineer with Newratek Inc. His research interests include molecular communications and next-generation 802.11 networks. He received the IEEE INFOCOM Best Demo Award, in 2015, and the IEEE WCNC Best Demo Award, in 2020.



MARTIN REISSLEIN (Fellow, IEEE) received the Ph.D. degree in systems engineering from the University of Pennsylvania, Philadelphia, PA, USA, in 1998. He is currently a Professor of electrical engineering and a Program Chair for computer engineering (CEN) with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, USA. He received the IEEE Communications Society Best Tutorial Paper Award, in 2008, the Friedrich

Wilhelm Bessel Research Award from the Alexander von Humboldt Foundation, in 2015, and the DRESDEN Senior Fellowship, in 2016 and 2019.



FRANK H. P. FITZEK (Senior Member, IEEE) received the Diploma (Dipl.-Ing.) degree in electrical engineering from the University of Technology Rheinisch-Westfälische Technische Hochschule, Aachen, Germany, in 1997, the Ph.D. (Dr.-Ing.) degree in electrical engineering from Technical University Berlin, Germany, in 2002, and the Honorary degree (Doctor Honoris Causa) from the Budapest University of Technology and Economics, Budapest, Hungary, in 2015. He was an

Adjunct Professor with the University of Ferrara, Italy, in 2002. In 2003, he joined Aalborg University as an Associate Professor, where he became a Professor. He co-founded several start-up companies, starting with Acticom GmbH, Berlin, in 1999. He is currently a Professor and the Head of the Deutsche Telekom Chair of Communication Networks, Technische Universität Dresden, Germany, where he coordinates the 5G Laboratory, Germany.



CHAN-BYOUNG CHAE (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from The University of Texas at Austin (UT), USA, in 2008.

Before joining UT, he was a Research Engineer with the Telecommunications Research and Development Center, Samsung Electronics, Suwon-si, South Korea, from 2001 to 2005. He is currently an Underwood Distinguished Professor with the School of Integrated Technology, Yonsei

University, South Korea. Before joining Yonsei University, he was with Bell Laboratories, Alcatel-Lucent, Murray Hill, NJ, USA, from 2009 to 2011, as a Member of the Technical Staff, and Harvard University, Cambridge, MA, USA, from 2008 to 2009, as a Postdoctoral Research Fellow.

Dr. Chae is a NAEK Fellow. He was a recipient/co-recipient of the Ministry of Education Award, in 2024; the KICS Haedong Scholar Award, in 2023; the CES Innovation Award, in 2023; the IEEE ICC Best Demo Award, in 2022; the IEEE WCNC Best Demo Award, in 2020; the Best Young Engineer Award from the National Academy of Engineering of Korea (NAEK), in 2019; the IEEE DySPAN Best Demo Award, in 2018; IEEE/KICS Journal of Communications and Networks Best Paper Award, in 2018; the IEEE INFOCOM Best Demo Award, in 2015; the IEIEE/IEEE Joint Award for Young IT Engineer of the Year, in 2014; the KICS Haedong Young Scholar Award, in 2013; the IEEE Signal Processing Magazine Best Paper Award, in 2013; the IEEE ComSoc AP Outstanding Young Researcher Award, in 2012; and the IEEE VTS Dan. E. Noble Fellowship Award, in 2008. He has held several editorial positions, including the Editor-in-Chief of IEEE TRANSACTIONS ON MOLECULAR, BIOLOGICAL, AND MULTI-SCALE COMMUNICATIONS, a Senior Editor of IEEE WIRELESS COMMUNICATIONS LETTERS, and a Editor of *IEEE Communications Magazine*, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and IEEE WIRELESS COMMUNICATIONS LETTERS. He was an IEEE ComSoc Distinguished Lecturer, from 2020 to 2023.

...