

Using Vector Fields for Efficient Simulation of Macroscopic Molecular Communication

Lukas Stratmann, Jan Peter Drees, Fabian Bronner, and Falko Dressler, *Fellow, IEEE*

Abstract—Molecular communication has been identified as a communication concept complementing radio communication in some areas and being the unique solution in others. This particularly includes communication between nano machines but, more recently, also macroscopic application domains such as in fluid systems in a chemical factory. We are interested in simulating such macroscopic molecular communication both accurately as well as on a large scale. In this work, we make use of the concept of vector fields for efficient simulation of particle movements in a fluid environment. Such vector fields can be pre-computed so that the simulation of the communication itself is very fast. We discuss both the concepts and the methodological approach to outline the advantages of this idea and validate the system compared to lab measurements. Going beyond previous work, we also integrated both on-off keying (OOK) and pulse position modulation (PPM) to demonstrate the feasibility of the simulation concept even for more complex signal processing tasks.

Index Terms—Molecular communication, vector fields, particle movement, modulation techniques

I. INTRODUCTION

Molecular communication (MC) has been intensively investigated in the last decade with a number of application scenarios [1], [2]. Initially motivated by the need for communication between nano systems, e.g., within the human body, there is now also a clear path for larger-scale application [1], [3]–[6]. It is envisioned that MC will even be integrated with future wireless networks, paving the road to 6G and beyond [5]. This research line is further supported by early standardization in the scope of the IEEE 1906 framework [7].

Even though MC was initially primarily considered for nano communication [1], recent research shows many promising applications in much larger-scale systems, now investigated as macroscopic MC. As one of the initial projects in this field, our Macroscopic Molecular Communication (MAMOKO) project¹ aims at paving the road for industrial applications, e.g., for communication in fluid environments in large chemistry factories.

Now, to support the development of systems, protocols, and tools for such industrial applications, a deeper fundamental understanding of the MC processes is needed. Channel modeling so far mainly focuses on diffusion processes [8], [9]. However, MC is strongly affected by the physical properties

of the carrier and the medium that transports it. Besides early experimentation, simulation is the fundamental basis for gaining more insights into the (macroscopic) MC processes because it has the potential for greater flexibility (e.g., in terms of the shape and scale of what types of channels can be used), lower costs (no specialized equipment or chemicals), and better time efficiency (e.g., by means of parallelization in large parameter studies).

Previous approaches, unfortunately, lack both granularity and scalability: Examples include NanoNS3 [10] (very low computation costs, exploration of higher-layer protocols such as routing), BiNS2 [11] (allowing to configure the environment more freely), and AcCoRD [12] (more computationally efficient mesoscopic domains). Furthermore, existing simulators tend to focus on environments where diffusion is the dominant cause of particle movement. While some simulators support uniform flows or, like BiNS2, even realistic analytically determined flow profiles in known objects like a straight tube, more complex geometries like obstacles or intersecting tubes cannot be realized easily in a flowing medium.

To fill this gap and enable the simulation of MC in complex flow profiles, our idea is to complement such diffusion- and uniform flow-based simulators using vector fields, which define a possibly different flow speed vector for every point in space. To this end, we recently proposed a workflow in which particles follow a vector field pre-computed with existing computational fluid dynamics (CFD) software [13]. In order to validate our system, we prepared a setup exactly matching a testbed experiment from Erlangen University [14].

In this paper, we go beyond our initial work [15] and also integrate modulation into the simulator. In particular, we integrate both on-off keying (OOK) and pulse position modulation (PPM) to demonstrate the feasibility for more complex MC systems on the physical layer. In a proof-of-concept study, we demonstrate the capabilities of our simulator and measure the bit error ratio over a macroscopic MC channel.

Our main contributions can be summarized as follows:

- We connect the concept of vector fields for simulating macroscopic MC with physical layer protocols,
- we integrate higher layer modulation techniques using OOK as well as PPM as a proof of concept, and
- we show results from initial simulations validated against testbed results.

II. RELATED WORK

A number of different simulation approaches for MC have been proposed in the past. For example, nanoNS3 is an extension of the popular discrete-event network simulator ns-3

L. Stratmann and F. Dressler are with the School of Electrical Engineering and Computer Science, TU Berlin, Germany, e-mail: {stratmann, dressler}@ccs-labs.org.

J. P. Drees is with the Bergische Universität Wuppertal, Germany, e-mail: jan.drees@uni-wuppertal.de.

F. Bronner is with the Dept. of Computer Science, Paderborn University, Germany, e-mail: fabian.bronner@ccs-labs.org.

¹<https://www2.tkn.tu-berlin.de/projects/mamoko/>

and was specifically developed with bacterial MC in mind [10]. It thus supports an efficient analytical channel model for the specific case of fluorescence-inducing bacteria in a microfluidic environment, which would have to be re-developed for different information carriers in macro-scale fluid channels. Other MC simulators support a more granular, particle-based simulation approach, which allows simulating the influence of particle collisions, outside forces such as magnetism, or uniform flow fields. Examples for such simulators include N3Sim [16], BiNS2 [11], and AcCoRD [12]. However, we noted previously that existing simulators specialized in diffusion-driven MC use simplified flow models [13]. To the best of our knowledge, simulating MC efficiently within a flowing medium is still an open problem if the flow profile cannot be determined analytically.

A known challenge of MC is intersymbol interference (ISI), which is exacerbated by residual particles left in the channel after each transmission, resulting in a memory effect [17]. Therefore, in order to transmit information efficiently, robust modulation and demodulation schemes are required. One solution to reduce ISI, for example, is to combine concentration shift keying (CSK) and molecule shift keying (MSK) by using different particle types to indicate whether a sequence of 1-valued bits continues or whether it transitions back to a 0-valued bit [18]. However, the requirement of MSK to be able to both emit and detect different types of particles is not always given, as the physical implementation of a transmitter or receiver typically becomes more complex if more particle types are to be supported. In these cases, OOK is a popular choice in diffusion-based MC. On the receiving end, it is often realized by comparing samples against a threshold [19]. While this approach is useful for validating testbeds or a simulator such as the one here presented due to its straight-forward implementation, there is a potential for improvement by opting for a thresholdless solution [20]. This is especially the case if residual particles that have accumulated over some time eventually exceed such thresholds on their own, even without any recent particle emissions. For the present study, however, we implemented the threshold-based OOK modulation scheme that was used for demonstrating the aforementioned testbed from Erlangen University [14]. This is explained in more detail in Section V.

III. THE POGONA SIMULATOR

We aim at simulating large-scale fluid-based MC. Thus, we need to assess both the flow characteristics of the fluid itself and the movement of MC particles within the fluid. To prepare a simulation, our approach takes the following steps.

(1) *Pre-Computation of Vector Fields* To generate a pre-computed vector field, first a geometry needs to be modeled in which the particles can later move. This may be a simple tube or more complex shapes such as a Y-piece as shown in Figure 1. For modeling, any CAD software capable of producing files in the STL format compatible with the CFD tool can be used. We choose the open-source tool Blender since we also use it for visualizing particle traces in a later step. Given a 3D model of the surface, we then use the OpenFOAM tool SnappyHexMesh

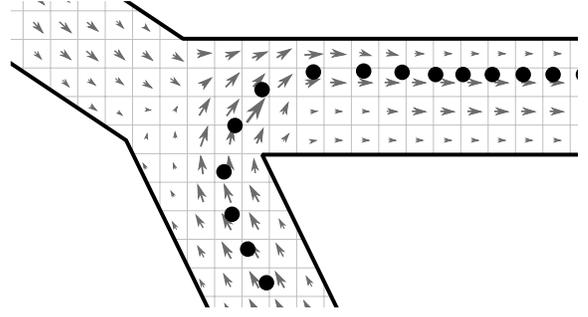


Figure 1. Conceptual trajectory of a simulated particle in a vector field model of a Y-piece. Background flow enters the tube from the left, the particle solution is injected from the bottom, and the resulting flow exits on the right.

to split this geometry into cells, which, in turn, can be used by OpenFOAM to find a numerical solution for the flow speed and pressure in each cell based on initial parameters for the inlets and outlets. Once this CFD simulation converges, we can export and use the final vector field of flow speeds in our MC simulator.

(2) *Flow Processing and Particle Simulation* As the next step, parameters for the MC simulation need to be defined, e.g., the number and rate of particles to inject, locations of sensors, and simulation duration. When the simulation runs, particle positions are updated and logged in discretized time steps based on the flow speeds of vector field cells in their vicinity. A typical simulation will have particles entering the scene in some sub-volume, for instance injected from a syringe or another tube, in which particles are spawned in a longitudinal section of the same tube. In principle, it would be possible to infer the number of simulated particles from the size of this sub-volume, an experimentally determined density of particles in the testbed, and a reduction factor. However, defining the injection flow rate and the number of particles independently as a more generalized implementation makes it easier, for example, to switch from a volume to a point injection. After instantiation, the particles are transported to and observed by a sensor, which can access information on all particles in its domain. At the sensor, typical communication metrics can be obtained, based on particle density and travel time from source to sensor. This way, we can easily measure, e.g., the channel impulse response or we can add additional higher-layer simulation models, e.g., to support modulation/demodulation of digital information on the particle flow. To achieve this, the modularity of the simulator allows adding new demodulation classes, for example, which can in turn access the sensor readings. Alternatively, demodulation scripts can use the sensor log files and process the output of the same simulation run in different variations after the fact. An example of the channel impulse response measured at such a sensor is shown in Figure 2.

To achieve acceptable computation times, we make some simplifying assumptions. In the testbed, several billion physical particles are involved in each injection. We therefore simulate only a small fraction of the real number of particles, though large enough for results that are sufficiently stable between simulation runs for a given analysis. We give these particles a uniformly distributed random starting position in an injector

volume. We update each particle position in discrete time steps. For adaptive step sizes and better integration accuracy in larger time steps we use the Runge-Kutta-Fehlberg (RK45) method [21].

CFD software commonly works on a volumetric three-dimensional mesh and produces outputs in the form of a vector field indicating the flow speed at discrete locations. A necessary discretization is therefore the division of the flow volume into cells. To accelerate the CFD pre-computation and vector field lookups, the mesh resolution should not be much higher than necessary. The mesh resolution can be kept lower by using appropriate interpolation methods for particle positions in the space between known flow speeds at the cell centers. We implemented a local variant of inverse distance weighting (IDW) [22], which means that we only need to look up the cells closest to a particle rather than iterating over the entire mesh.

We improve computation efficiency further by using a k-d tree for vector field lookups in, on average, $\mathcal{O}(\log n)$ in a spatial data structure [23]. Additionally, we introduce a scene system for connecting and reusing multiple vector fields. If the same geometry appears multiple times in one scene with matching flow rates, the number of expensive CFD pre-computations in larger-scale networks can be kept low. Further efficiency can be gained when assuming stationary sensors, in which case sensors can subscribe to individual vector field cells and avoid being addressed by each particle in each time step [15]. Sensor subscriptions are initialized in a pre-processing step before the actual simulation starts, the result of which is a list of references to all relevant sensors for each vector field cell. Instead of iterating over all sensors, this way, each particle only needs to determine its closest cell by means of a k-d tree lookup, after which the list of relevant sensors can be determined in constant time.

We strive to reduce mistakes in the process of defining simulation scenarios and to allow easy inspection of particle traces within the scene. This is especially critical in complex scenes in which many vector field objects and components like sensors need to align precisely. We achieve this with an optional Blender add-on which can export an arrangement of simulation components to a configuration file, and which can import simulated particle traces for visualization.

IV. VALIDATION AGAINST TESTBED DATA

We compared the full width at half maximum (FWHM) and root mean square (RMS) delay spreads of the channel impulse response after 5 cm in our simulator with data from a testbed [14]. For the result shown in Figure 2, we simulated 16 independent Y-piece injections of 5000 particles each. 26 μL of particle solution were injected at a rate of 10 mL/min into a background flow of 5 mL/min. In total, the scenario consisted of 1 292 703 vector field cells (646 883 for the Y-piece, 327 910 for a connected 15 cm long tube 2.5 cm after the point of injection, and again as many for the same tube connected to the particle inlet). The tube radius was 0.76 mm and the sensor was placed with its center at 5 cm after the point of injection. On average, the simulator showed a FWHM

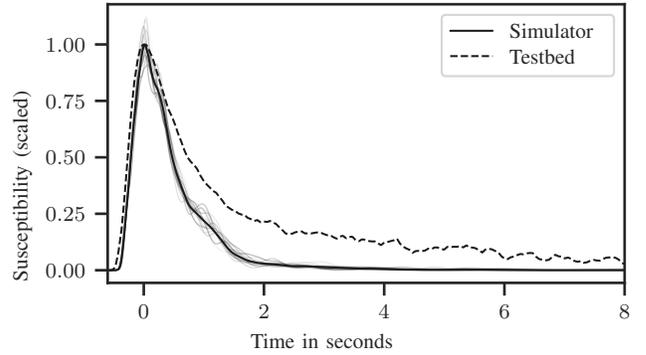


Figure 2. Average channel impulse response from 16 Y-piece injections of 5000 particles compared to testbed data [14]. Individual simulation runs are shown in gray. Susceptibility values are scaled according to the respective maximum of the average channel impulse responses.

of 689.6 ms (standard deviation (SD) = 31.3 ms), which is 81.8% of that of the testbed at 842.8 ms (40 injections, SD = 100.0 ms). The RMS delay spreads, however, are as yet significantly underestimated by a factor of 5, with an average of 0.439 s (SD = 0.024 s) in the simulator compared to 1.874 s (SD = 0.409 s) in the testbed. This is in line with our previous results for a comparison of testbed and simulation data over different channel lengths [15]. These differences may be caused by several physical phenomena which we do not yet simulate, like turbulence at the point of injection, diffusion, or sedimentation. However, the channel impulse responses between simulation and testbed are arguably already similar enough to motivate using this approach for simulating communication.

V. INTEGRATING OOK AND PPM MODULATION

In a next step, we demonstrate the possibility to use our simulator for simulating the transmission of binary information. As a proof of concept, we replicated a testbed experiment in which on-off keying (OOK) was used to transmit the letters ‘FAU’ [14]. We re-used the scenario of the channel impulse response experiment above, but this time with an injection rate of 5 mL/min and an injection volume of 14 μL to more closely match the original OOK experiment. We configured 2000 particles to be spawned for each injection. The symbol duration was 4 s and each character of the letter sequence was ASCII-encoded in respective sequences of 8 bits, whereas the first 3 bits of the upper-case ASCII alphabet (010) were used for synchronization. More specifically, the preamble is detected if a predetermined threshold $c = \frac{1}{3}$ is exceeded, which approximately corresponds to the threshold used in the testbed [14]. Afterwards, the greatest-valued sample within the next half of a symbol duration is taken to be the peak of the preamble. Subsequently, the signal is sampled in intervals of the symbol duration after the initial preamble peak. If the sampled susceptibility is above c , the corresponding bit is decoded as 1, otherwise it is 0. Figure 3 shows a successful transmission using this method for a symbol duration of 4 s. The long symbol duration and comparatively short injection times make the peaks easily discernible, resulting in a reliable transmission of the letter sequence.

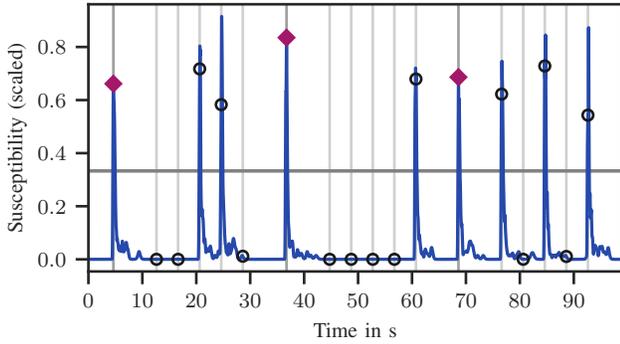


Figure 3. Proof-of-concept OOK transmission of the letters ‘FAU’ with a symbol duration of 4 s. Diamonds indicate the detected peak of a 010 preamble. Circles indicate the sampled values for each payload bit. The threshold is indicated by a grey horizontal line.

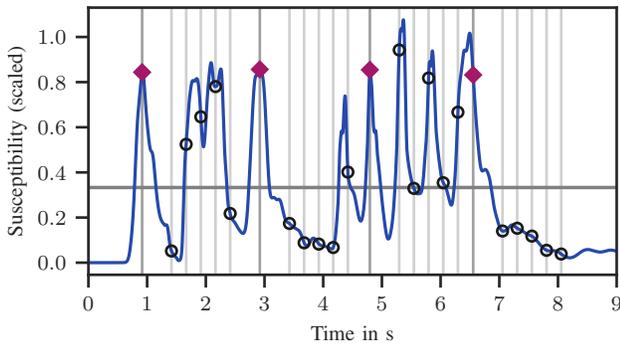


Figure 4. OOK with a symbol duration of 0.25 s.
Received data: 01001110 01000001 01010111 01000000,
expected data: 01000110 01000001 01010101.

As ISI is known to be an important constraint in MC, we next sought insights into the robustness of the modulation scheme and how it compares to PPM. Illustrating the effect of ISI, Figure 4 shows an unsuccessful transmission of the letter sequence ‘FAU’ for a symbol duration of 0.25 s. Especially when many 1-valued bits are transmitted in close succession, it can be seen that susceptibility levels barely have time to fall back below the threshold. After 4 s, peaks appear to grow successively higher as new and residual particles combine.

PPM is another modulation scheme that has been proposed for diffusive [24] and macroscopic [25] MC. For M -ary PPM, the duration of one symbol is divided into M chips of equal duration. To transmit a symbol of value m , a pulse is generated in the m -th chip. In contrast to OOK, for which we expect at most one peak per symbol, PPM therefore guarantees exactly one transmitted pulse for every symbol. The receiver uses the maximum likelihood (ML) decoding technique as described, e.g., in [24]: For each chip of a symbol, the detected numbers of particles in each sample are accumulated. Since we simulate the response of a susceptibility sensor rather than counting particles [15], we accumulate the susceptibility samples. Then we compare the M resulting values and decode as the symbol value the index of the chip with the maximum value.

We see the ISI effects described above for Figure 4 also in Figure 5, where we transmitted random bit sequences to obtain

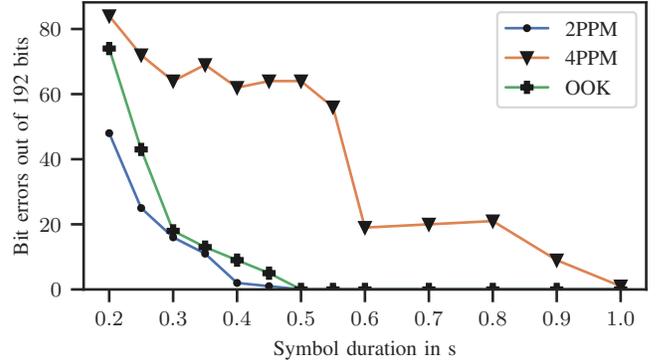


Figure 5. Bit error ratios for different symbol durations.

the bit error ratio (BER) for different symbol durations. In order to make the modulation schemes OOK, 2-ary, and 4-ary PPM comparable, we disabled the synchronization technique that was used in the testbed for OOK and instead synchronize only on the first detected peak in all modulation schemes. For each tested symbol duration and modulation scheme we ran 4 simulations and in each we transmitted 48 random bits. On an AMD Ryzen™ 7 3800X CPU, one single-threaded simulation run of this setup in our current Python implementation took about 7 h in the case of OOK.

We observe consistently fewer bit errors with 2-ary PPM compared to OOK, which in turn produced significantly fewer bit errors than 4-ary PPM in our simulations. At the lowest symbol duration of 0.2 s, 2-ary PPM produced 48, OOK 74, and 4-ary PPM produced 84 bit errors out of 192 transmitted bits. This corresponds to BERs of 25 %, 39 %, and 44 %, respectively. We did not investigate lower symbol durations in this case, as the injection duration for the given injection rate and volume was 0.17 s. For a symbol duration of 0.45 s, we observed only one bit error with 2-ary PPM and 5 with OOK. For symbol durations beyond 0.5 s, we did not see any bit errors with either modulation scheme. 4-ary PPM, however, only achieves BERs below 5 % with a symbol duration of 0.9 s and one bit error out of 192 for 1 s. The improvement we see in 2-ary PPM over our implementation of OOK may for a large part be due to the decoding technique. While OOK here relies on the detected susceptibility to be either high or low enough in one specific sample for each symbol, PPM takes the aggregate of multiple samples into account. However, as this improvement is consistent but small, OOK may still be the preferred choice in some applications of MC due to its lower expected number of transmitted pulses and thereby fewer used-up particles. The low performance of 4-ary PPM can be explained by the higher precision that is required in both the shape of the peaks and time synchronization, due to the greater subdivision of the same symbol durations.

VI. CONCLUSION

Simulation of macroscopic MC needs fine-grained yet scalable models and methods to evaluate and better understand communication effects on the physical layer (and above). Using a fluid-based MC channel as an example, we investigated the use of vector fields for efficient simulation of such macroscopic

MC. We suggest the use of pre-computed vector fields to substantially speed up the simulation of particle flow in a fluid environment. Validating our system against a wet-lab testbed, we show that our approach can achieve a very high accuracy. Further investigations are required to determine under which conditions our approach is feasible and in which cases additional physical phenomena such as turbulence, diffusion, or gravity have to be considered.

We furthermore showed that our system can simulate macro-scale OOK and PPM transmissions while realistically recreating real-world effects like ISI due to residual particles. In future work, this may be useful in exploring the efficacy of communication schemes originally developed for nano-scale MC, for the development of new schemes, i.e., novel types of transmitter particles or novel modulation techniques, or for investigating higher-layer protocols in complex fluid networks.

ACKNOWLEDGMENT

Reported research was supported in part by the project *MAMOKO* funded by the German Federal Ministry of Education and Research (BMBF) under grant number 16KIS0917.

REFERENCES

- [1] I. F. Akyildiz, J. M. Jornet, and M. Pierobon, "Nanonetworks: a new frontier in communications," *Communications of the ACM*, vol. 54, no. 11, pp. 84–89, Nov. 2011.
- [2] S. F. Bush, J. L. Paluh, G. Piro, V. Rao, V. Prasad, and A. W. Eckford, "Defining Communication at the Bottom," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 1, no. 1, pp. 90–96, Mar. 2015.
- [3] F. Dressler and S. Fischer, "Connecting In-Body Nano Communication with Body Area Networks: Challenges and Opportunities of the Internet of Nano Things," *Elsevier Nano Communication Networks*, vol. 6, pp. 29–38, Jun. 2015.
- [4] N. Farsad, H. B. Yilmaz, A. W. Eckford, C.-B. Chae, and W. Guo, "A Comprehensive Survey of Recent Advancements in Molecular Communication," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1887–1919, 2016.
- [5] W. Haselmayr, A. Springer, G. Fischer, C. Alexiou, H. Boche, P. A. Hoehner, F. Dressler, and R. Schober, "Integration of Molecular Communications into Future Generation Wireless Networks," in *1st 6G Wireless Summit*. Levi, Finland: IEEE, Mar. 2019.
- [6] M. Kuscü, E. Dinc, B. A. Bilgin, H. Ramezani, and O. B. Akan, "Transmitter and Receiver Architectures for Molecular Communications: A Survey on Physical Design With Modulation, Coding, and Detection Techniques," *Proceedings of the IEEE*, vol. 107, no. 7, pp. 1302–1341, Jul. 2019.
- [7] "Recommended Practice for Nanoscale and Molecular Communication Framework," IEEE, Std 1906.1-2015, Jan. 2016.
- [8] T. Khan, B. A. Bilgin, and O. B. Akan, "Diffusion-Based Model for Synaptic Molecular Communication Channel," *IEEE Transactions on NanoBioscience*, vol. 16, no. 4, pp. 299–308, Jun. 2017.
- [9] V. Jamali, A. Ahmadzadeh, W. Wicke, A. Noel, and R. Schober, "Channel Modeling for Diffusive Molecular Communication—A Tutorial Review," *Proceedings of the IEEE*, vol. 107, no. 7, pp. 1256–1301, Jul. 2019.
- [10] Y. Jian, B. Krishnaswamy, C. M. Austin, A. O. Bicen, J. E. Perdomo, S. C. Patel, I. F. Akyildiz, C. R. Forest, and R. Sivakumar, "nanoNS3: Simulating Bacterial Molecular Communication Based Nanonetworks in Network Simulator 3," in *3rd ACM International Conference on Nanoscale Computing and Communication (NANOCOM 2016)*. New York City, NY: ACM, Sep. 2016, pp. 17:1–17:7.
- [11] L. Felicetti, M. Femminella, and G. Reali, "Simulation of molecular signaling in blood vessels: Software design and application to atherogenesis," *Elsevier Nano Communication Networks*, vol. 4, no. 3, pp. 98–119, Sep. 2013.
- [12] A. Noel, K. C. Cheung, R. Schober, D. Makrakis, and A. Hafid, "Simulating with AcCoRD: Actor-based Communication via Reaction–Diffusion," *Elsevier Nano Communication Networks*, vol. 11, pp. 44–75, Mar. 2017.
- [13] F. Bronner and F. Dressler, "Towards Mastering Complex Particle Movement and Tracking in Molecular Communication Simulation," in *6th ACM International Conference on Nanoscale Computing and Communication (NANOCOM 2019), Poster Session*. Dublin, Ireland: ACM, Sep. 2019, pp. 36:1–36:2.
- [14] H. Unterweger, J. Kirchner, W. Wicke, A. Ahmadzadeh, D. Ahmed, V. Jamali, C. Alexiou, G. Fischer, and R. Schober, "Experimental Molecular Communication Testbed Based on Magnetic Nanoparticles in Duct Flow," in *19th IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2018)*, Kalamata, Greece, Jun. 2018, pp. 1–5.
- [15] J. P. Drees, L. Stratmann, F. Bronner, M. Bartunik, J. Kirchner, H. Unterweger, and F. Dressler, "Efficient Simulation of Macroscopic Molecular Communication: The Pogona Simulator," in *7th ACM International Conference on Nanoscale Computing and Communication (NANOCOM 2020)*. Virtual Conference: ACM, Sep. 2020.
- [16] I. Llatser, D. Demiray, A. Cabellos-Aparicio, D. T. Altılar, and E. Alarcón, "Simulation framework for diffusion-based molecular communication nanonetworks," *Simulation Modelling Practice and Theory*, vol. 42, pp. 210–222, Mar. 2014.
- [17] N. Farsad, H. B. Yilmaz, A. W. Eckford, C.-B. Chae, and W. Guo, "A Comprehensive Survey of Recent Advancements in Molecular Communication," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1887–1919, 2016.
- [18] B. Tepekule, A. E. Pusane, H. B. Yilmaz, and T. Tugcu, "Energy Efficient ISI Mitigation for Communication via Diffusion," in *IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, Odessa, Ukraine, May 2014, pp. 33–37.
- [19] M. Ş. Kuran, H. B. Yilmaz, T. Tugcu, and I. F. Akyildiz, "Modulation Techniques for Communication via Diffusion in Nanonetworks," in *IEEE International Conference on Communications (ICC 2011)*, Kyoto, Japan, Jun. 2011.
- [20] S. Sharma, K. Deka, and V. Bhatia, "Thresholdless Detection of Symbols in Nano-Communication Systems," *IEEE Transactions on NanoBioscience*, vol. 19, no. 2, pp. 259–266, Apr. 2020.
- [21] E. Fehlberg, "Low-order classical Runge-Kutta formulas with stepsize control and their application to some heat transfer problems," George C. Marshall Space Flight Center, National Aeronautics and Space Administration, Tech. Rep., Jul. 1969. [Online]. Available: <http://hdl.handle.net/2060/19690021375>
- [22] R. Franke and G. Nielson, "Smooth interpolation of large sets of scattered data," *International Journal for Numerical Methods in Engineering*, vol. 15, no. 11, pp. 1691–1704, 1980.
- [23] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, Sep. 1975.
- [24] A. Zare, A. Jamshidi, and A. Keshavarz-Haddad, "Receiver Design for Pulse Position Modulation Technique in Diffusion-Based Molecular Communication," in *4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)*. Tehran, Iran: IEEE, Dec. 2017, pp. 729–733.
- [25] S. Bhattacharjee, M. Damrath, and P. A. Hoehner, "Design of macroscopic air-based molecular communication concept using fluorescein," in *7th ACM International Conference on Nanoscale Computing and Communication (NANOCOM 2020), Poster Session*. Virtual Conference: Association for Computing Machinery (ACM), 9 2020.