

Simulating with AcCoRD: Actor-Based Communication via Reaction-Diffusion

Adam Noel^a, Karen C. Cheung^b, Robert Schober^c, Dimitrios Makrakis^a, Abdelhakim Hafid^d

^a*School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada*

^b*Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada*

^c*Institute for Digital Communication, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany*

^d*Department of Computer Science and Operations Research, Université de Montréal, Montréal, QC, Canada*

Abstract

This paper introduces AcCoRD (Actor-based Communication via Reaction-Diffusion) version 1.0. AcCoRD is a sandbox reaction-diffusion solver designed for the study of molecular communication systems. It uses a hybrid of microscopic and mesoscopic simulation models that enables scalability via user control of local accuracy. AcCoRD is developed in C as an open source command line tool and includes utilities to process simulation output in MATLAB. The latest code and links to user documentation can be found at <https://github.com/adamjnoel/AcCoRD/>. This paper provides an overview of AcCoRD's design, including the motivation for developing a specialized reaction-diffusion solver. The corresponding algorithms are presented in detail, including the computational complexity of the microscopic and mesoscopic models. Other novel derivations include the transition rates between adjacent mesoscopic subvolumes of different sizes. Simulation results demonstrate the use of AcCoRD as both an accurate reaction-diffusion solver and one that is catered to the analysis of molecular communication systems. A link is included to videos that demonstrate many of the simulated scenarios. Additional insights from the simulation results include the selection of suitable hybrid model parameters, the impact of reactive surfaces that are in the proximity of a hybrid interface, and the size of a bounded environment that is necessary to assume that it is unbounded. The development of AcCoRD is ongoing, so its future direction is also discussed in order to highlight improvements that will expand its potential areas of application. New features that are being planned at the time of writing include a fluid flow model and more complex actor behavior.

Keywords: molecular communication, reaction-diffusion, microscopic simulation, mesoscopic simulation

1. Introduction

1.1. Background and Motivation

There has been recent interest in designing synthetic wireless communication networks for environments where conventional (i.e., radio frequency) wireless technologies are unsafe, infeasible, or impractical, such as in biological systems. This interest has inspired the idea of adapting natural communication strategies from these biological systems. One such strategy is molecular communication (MC), initially proposed for synthetic networks in [1], where transmitters use physical molecules as information carriers. MC is used for signaling in nature over a wide range of physical scales, from quorum sensing in bacterial communities to communication via pheromones over a kilometer or more, as described in [2] and [3, Ch. 53], respectively. In particular, MC is ubiquitous in communication within and between cells; see [4, Ch. 16].

Natural MC systems are typically designed for the transmission of limited quantities of information, such as

a time-varying ON/OFF control signal for some process. However, synthetic MC networks are envisioned for transmitting arbitrarily large amounts of information. These networks could enable new applications in fields including biological engineering, medicine, manufacturing, and environmental modeling; see [5].

Commonly-studied forms of MC include molecular *diffusion*, where molecules passively propagate via collisions with other molecules in a fluid environment. The interest in diffusion for communication engineering, as demonstrated in [6], can be attributed to its speed over very short distances (particularly on the scale of a micron or less), its simplicity (requiring no active propagation mechanism), and the availability of mathematical models to facilitate analysis. In particular, mathematical models are needed to determine a channel's impulse response (i.e., the time-varying signal observed at a receiver due to the release of an instantaneous signal by a transmitter). Knowledge of the channel impulse response is essential for meaningful transceiver system design and performance analysis.

There are many seminal texts on the analysis of diffusion, e.g., [7, 8, 9]. However, closed-form expressions for the impulse response of a diffusive channel generally require simplifying assumptions and specific system geome-

Email addresses: anoel2@uottawa.ca (Adam Noel), kcheung@ece.ubc.ca (Karen C. Cheung), robert.schober@fau.de (Robert Schober), dimitris@eecs.uottawa.ca (Dimitrios Makrakis), ahafid@iro.umontreal.ca (Abdelhakim Hafid)

tries. This reality is an obstacle to the development of MC networks. Most existing research has considered a variation of a common system model. Typically, authors will consider a one- or three-dimensional (i.e., 1D or 3D) unbounded environment, possibly with a uniform fluid flow, with a point source and a receiver that is either an absorbing surface (i.e., molecules are “consumed”) or a passive observer¹.

While the resulting analysis of simplified models is convenient, and uniformity is helpful for comparing transceiver designs, there are a couple of issues with this trend. First, realistic diffusive environments are generally bounded. Approximating an environment as unbounded is only appropriate if it is symmetric and in the absence of other local obstacles, but environments such as cells and cellular tissues are full of obstacles; see [4, Ch. 20]. Second, diffusion is not the only phenomenon that can affect the behavior of a diffusing particle. In addition to fluid flow (which is in general not uniform; see [10]), molecules can undergo chemical reactions (besides absorption at a surface) that convert them into other molecular species or transport them across boundaries. For example, see [11, Ch. 9] for elementary analysis of chemical reactions.

Flow and reactions can significantly modify the channel response, and could even be deliberately introduced to improve communication performance. This was observed by using an electric fan in the macroscale testbench developed in [12], and by adding enzymes to the propagation environment in our own work in [13]. Neither of these strategies could be accurately described using the commonly-studied channel models; the experiments in [12] were fitted to a corrected 1D model in [14], and the enzyme kinetics in [13] were simplified to a first order degradation reaction so that the impulse response could be derived.

The notion that we simplified the analytical model in [13] immediately begs the question of how we verified its accuracy. Furthermore, it was insufficient to determine the *expected* channel behavior; for communications analysis, we are interested in the *probability distribution* of the channel behavior. We addressed both issues in [13] by simulating the detailed model. Unfortunately (and to the best of our knowledge), no existing simulation platform would accommodate the detailed model in [13]. Even though we could have used a *generic reaction-diffusion solver* such as Smoldyn (see [15]) to evaluate the expected channel behavior, it was not suitable for assessing the time-varying channel statistics or for configuring a source to release molecules based on modulating a sequence of random binary data. Our solution in [13] was to internally develop a simulator in MATLAB. However, this simulator was specific to the environment of the model in [13] and not easily portable to other system models. We claim that existing publicly-available MC simulators have similar limitations,

¹A (perhaps surprisingly) large fraction of papers reviewed in [6], including the current authors’ own work on MC, can be classified as using such a model.

e.g. [16, 17, 18, 19, 20] (which we will discuss in further detail in Section 2.2).

1.2. The AcCoRD Simulator

With these limitations in mind, we have developed the AcCoRD simulator (Actor-based Communication via Reaction-Diffusion). AcCoRD is a generic reaction-diffusion solver that is developed in C and designed for communications analysis. It is an open source project in active development on Github; see [21]. As of the time of writing (November 2016), the latest release is version 1.0. It has the following primary features:

- AcCoRD is a hybrid solver that integrates two simulation models to define 3D environments with flexible local accuracy. Each local region is classified as *microscopic* or *mesoscopic*. Microscopic regions define each molecule individually and evolve over discrete time steps. Mesoscopic regions count the number of molecules in disjoint virtual bins (called *subvolumes*) and evolve over time steps with continuous granularity. We increase the scalability by accommodating the placement of adjacent mesoscopic regions that have subvolumes of different sizes. This feature is based on an extension of the 2D system that we proposed in [22].
- Actors can be distributed throughout the environment as *active* molecule sources (i.e., transmitters) or *passive* observers (i.e., receivers). Transmitters release molecules according to the modulation of a random binary data sequence. The precise number of molecules released and the release times for a given symbol interval can be deterministic or randomized. Receivers can record the number of molecules and (optionally) their positions at any specified interval. Future development will couple these two actor classes to enable transceivers that behave according to their observations.
- Zeroth, first, and second order chemical reactions can be defined locally or globally, i.e., over the entire propagation environment or in a particular set of regions. This general framework can accommodate reactions such as molecule degradation, enzyme kinetics, reversible or irreversible surface binding, ligand-receptor binding, transitions across boundary membranes, and simplified molecular crowding. Surface binding reactions include absorption, i.e., *consumption*, adsorption, i.e., *sticking*, and desorption, i.e., *release* from a surface. Generally, we will refer to adsorption as reversible absorption.
- AcCoRD implements some microscopic behavior in continuous time. Specifically, the release of molecules by active actors, zeroth order reactions, and most first order reactions can occur at any time. Thus, a molecule can undergo multiple reactions in

a single microscopic time step, and the accuracy of these phenomena are independent of the chosen time step.

- Independent realizations of a simulation can be repeated an arbitrary number of times (and on different computers) and then aggregated to determine the average behavior and channel statistics.
- The online documentation includes installation and usage instructions for the latest version, descriptions of all configuration options, and many sample configuration files; see [21]. The sample configurations are provided to demonstrate all of AcCoRD’s functionality.
- AcCoRD’s interface has been designed to be helpful to novice users by providing descriptive output messages. AcCoRD also includes post-processing tools developed in MATLAB. These tools enable the aggregation of simulation output files, plotting receiver observations (either the time-varying behavior or empirical distributions at specified times), and visualizing the physical environment (either as still images or compiled into a video²).

Four sample environments that illustrate AcCoRD’s primary features are shown in Figs. 1 to 4. Figs. 1 and 2 show hybrid environments with both microscopic and mesoscopic regions. Fig. 3 has reversible surface reactions. Fig. 4 has a distinct transmitter and receiver. More details about these environments, including their simulation results, are discussed in Section 7.

AcCoRD provides significant flexibility for a user to define the physical environment, specify chemical reactions, and place sources and observers of molecules. We describe AcCoRD as a reaction-diffusion “sandbox”, i.e., this flexibility enables users to explore (“play”) and create their own system. As a communications analysis tool, it can model data modulation at transmitters and generate channel statistics at observers by repeating a simulation an arbitrary number of times. By developing AcCoRD as a “sandbox” reaction-diffusion solver from the perspective of molecular communication, we anticipate that it will contribute the following for the MC community:

- Encourage the use of simulations. Most authors in this domain have thus far verified their work via numerical evaluations of their analysis or via Monte Carlo simulations. In the latter case, the time-varying probability distribution of the channel behavior is assumed to be known. By omitting simulations, assumptions about the model and their accuracy for different parameter values might be untested.

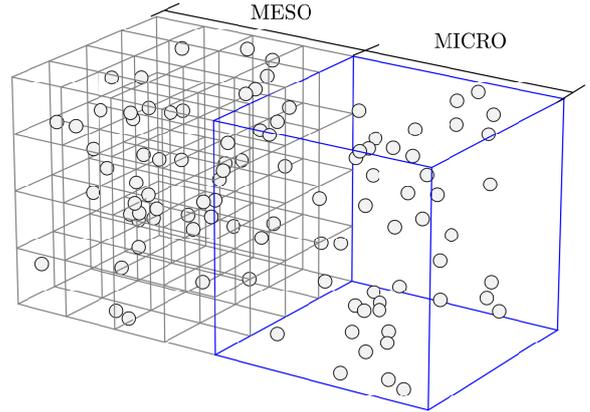


Figure 1: Simulation snapshot of a hybrid environment where the diffusing molecules are initially uniformly distributed. The left half is *mesoscopic*, where molecules are counted in one of 64 cubes (i.e., *subvolumes*), and the right half is *microscopic*, where molecules are tracked individually. 100 diffusing molecules are displayed. This environment is labeled System 1 and variations of it are simulated in Section 7.1. A sample video of this simulation is available as Video 2 in [23].

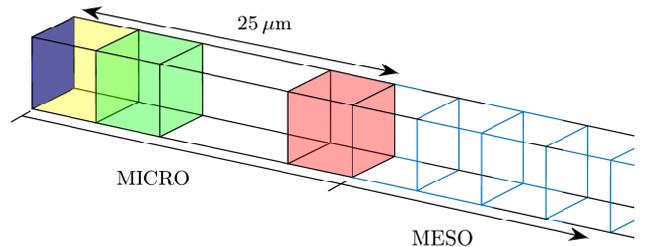


Figure 2: A portion of a hybrid environment with a reactive surface. The environment is a 0.5 mm long rod. At one end (shown in dark blue) is an absorbing surface. Three observers watch the molecules present over 5 μm sections of the rod (shown in yellow, green, and red). This environment is labeled System 2 and variations of it are simulated in Section 7.2. The variation shown has a hybrid interface between microscopic and mesoscopic regions at a distance of 25 μm from the absorbing end, such that the cubes with blue outlines are mesoscopic subvolumes. A sample video of the simulation of this environment is available as Video 3 in [23].

- Improve the understanding and visualization of known reaction-diffusion environments and their channel responses. This is especially important for improving the accessibility of this multi-disciplinary field and encouraging new researchers to contribute without needing to develop their own simulation tools.
- Provide a platform to verify new analysis and test transceiver designs. For example, we used AcCoRD in [25] to study the accuracy of the common assump-

²A series of eight videos are discussed throughout this paper and can be found at <https://www.youtube.com/playlist?list=PLZ7uYXG-7XF8UyhFrIuQIiZig1XA89e3i>; see [23].

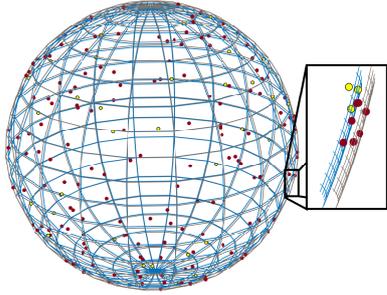


Figure 3: Simulation snapshot of an environment with two large concentric spheres with radii $120\ \mu\text{m}$ and $122\ \mu\text{m}$. Molecules are initialized in the space between the two spheres. This environment is a spherical analog to the 1D system studied in [24, Fig. 6a]. The inner and outer spherical surfaces are shown with blue and grey outlines, respectively. This environment is labeled System 3 and variations of it are simulated in Section 7.3. In the variation shown, molecules can probabilistically diffuse through the inner surface, i.e., the inner surface acts as a membrane. The molecules turn from red to yellow when they transition inside the inner sphere (see inset). Sample videos of this simulation are available as Video 4 (for absorption to the outer sphere) and Video 5 (for transitions through the membrane) in [23].

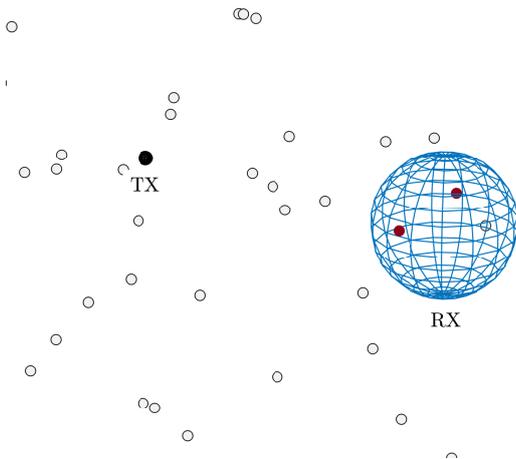


Figure 4: Simulation snapshot of a simple communication environment. Molecules are released by a point transmitter and observed at a spherical receiver (radius $1\ \mu\text{m}$) centered $5\ \mu\text{m}$ from the transmitter. This environment is labeled System 4 and variations of it are simulated in Section 7.4. In the variation shown, the released molecules (light grey) can irreversibly bind to the receiver’s absorbing surface (and turn red when bound). One molecule that appears to be inside the receiver is actually behind it. A sample video of this simulation is available as Video 8 in [23]. Other variations of this system are shown in Videos 6 and 7 in [23].

tion that the molecule source is a point transmitter.

- Enable the exploration of new channels that have not or cannot be precisely examined analytically, such as the diffusion model with enzyme kinetics that we considered in [13].

1.3. Contributions

This paper introduces the AcCoRD simulator, and also makes the following novel technical contributions:

1. We derive the transition rate between adjacent 3D mesoscopic subvolumes of different sizes that have partially-overlapping faces. The 2D version of this rate was presented in [22]. Accommodating mesoscopic subvolumes with different sizes increases the flexibility of local accuracy.
2. We derive the continuous event time for a first order microscopic chemical reaction event that is “known” to have occurred in the current microscopic time step. This derivation is inspired by how chemical reactions occur in both the microscopic and mesoscopic models, and enables a molecule to react multiple times within a single microscopic time step (albeit with an increased computational cost if multiple reactions do occur).
3. We verify AcCoRD’s accuracy by comparing its simulation output with analytical results for known reaction-diffusion environments. Examples include surface reactions, enzyme kinetics, and common molecular communication channels. Early versions of some of these results in 2D environments were presented in [26], but no specific simulations are repeated here.
4. We demonstrate scalability by investigating the trade-off between computational efficiency and local accuracy in hybrid and multi-scale environments. We provide insights into the appropriate size and placement of subvolumes in the mesoscopic regime, the impact of a reactive surface near the proximity of the hybrid interface, and make general comments on when a hybrid model is appropriate. Preliminary versions of some of these results with simpler hybrid transition rules and in 2D environments were presented in [22, 26].
5. We demonstrate that a bounded environment can be treated with an unbounded model if the distance from the area of interest to the nearest boundary is at least three times the average diffusion distance for the time scale of interest.

1.4. Organization

The rest of this paper is organized as follows. We review related simulation tools, including generic reaction-diffusion solvers and diffusive MC simulators, in Section 2. In Section 3, we describe the components of an AcCoRD simulation. Section 4 presents the underlying theory for reaction and diffusion behavior. It includes our derivation for the transition rate between mesoscopic subvolumes

of different sizes and for the reaction times of first order chemical reactions that have occurred within a microscopic time step. We present the overall AcCoRD algorithm and discuss computational complexity in Section 5. We summarize the interface and work flow for using AcCoRD in Section 6. In Section 7, we present detailed simulation results to demonstrate AcCoRD’s functionality and to verify its accuracy by comparing with analytical results. In Section 8, we identify features for AcCoRD’s future development. Conclusions are drawn in Section 9.

The Appendices focus on additional technical details that we include for completeness. Appendix A lists constraints and limitations to defining simulation environments. Appendix B describes the surface reaction probabilities that were mostly derived in [24] but are also implemented in AcCoRD. In Appendix C, we present the detailed algorithms of the main simulation steps. We review probability distributions and the calculation of mutual information in Appendix D. Furthermore, sample simulation videos of many of the environments studied in this paper are included in [23] and summarized in Appendix E.

A reader who is only interested in using the AcCoRD software can focus on Sections 3 and 6. A reader who is interested in the details of the implementation should refer to Sections 3, 4, 5, Appendix B, and Appendix C.

2. Related Work

In this section, we summarize existing simulation tools that implement features comparable to those developed in AcCoRD. First, we present a general discussion of reaction-diffusion solvers, including the simulation models that are commonly used over various physical scales. We also discuss efforts to develop simulations that integrate multiple models for improved scalability. Then, we focus on describing the simulation tools that have been developed specifically for the study of diffusive molecular communication systems. Finally, we draw comparisons between AcCoRD and existing tools.

2.1. Reaction-Diffusion Solvers

Generic reaction-diffusion solvers make trade-offs between simulation accuracy and computational efficiency to model molecule behavior. As we discussed in [26], solver models can usually be placed into one of four categories that correspond to the degree of local detail of molecular behavior. In order of increasing computational efficiency (and therefore decreasing accuracy), these categories are *molecular dynamics* models, *microscopic* models, *mesoscopic* models, and *continuum* models. Here, we discuss each of these categories in sequence, and then describe efforts to implement simulation models over multiple scales.

2.1.1. Molecular Dynamics

The most detailed solvers use *molecular dynamics* and account for all interactions between all individual molecules, including fluid solvents (e.g., water). A classical molecular dynamics simulator is the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS); see [27] and details of ongoing development in [28]. For example, LAMMPS can model rigid body particles, charge potentials, and electron force fields, where the orientations of individual particles are accounted for.

2.1.2. Microscopic Models

The next most-detailed solvers use *microscopic* models which treat solvent molecules as a continuum but all other molecules are individually tracked. Modeling the solvent as a continuum enables the introduction of a diffusion coefficient to describe the motion of an individual molecule as it moves within the solvent; see [8]. One microscopic solver is the Smoluchowski Dynamics simulator (Smoldyn); see [15, 29]. Smoldyn uses a constant time step and tracks the number and state of molecules over subsequent time steps. An alternative to this approach is Green’s-function reaction dynamics (GFRD), which uses a continuous time model and tracks changes in molecules’ states (e.g., when two molecules collide and react); see [30].

2.1.3. Mesoscopic Models

More computationally efficient solvers do not model individual molecules but partition the simulation environment into virtual containers. This approach becomes more efficient as the environmental complexity grows (e.g., with the presence of more molecules). The third category of solvers uses *mesoscopic* models, which refer to the containers as “subvolumes” and track discrete numbers of molecules in each container. Reactions involving two or more molecules can only occur if all reactants are in the same subvolume. Chemical reactions occur as “events”, and diffusion is modeled as an event where a molecule moves from one subvolume to another. In their most accurate form, mesoscopic models execute one reaction at a time, and this approach can capture system behavior *exactly* (in a statistical sense) if molecule populations throughout each subvolume are homogeneous and the size of each subvolume is chosen in consideration of the time scales of potential events; see [31].

There are a few ways to increase computational efficiency via scalability in mesoscopic models. One method is by adjusting the size and placement of the subvolumes themselves. Examples of this method use subvolumes that are tetrahedrons (as in the URDME simulator in [32]) or subvolumes whose size can change in both time and space (as considered for square subvolumes in [33]). Another method is to evolve the system according to time steps and execute multiple events in each step. This approach is known as “tau-leaping”; see [34] for its introduction in reaction-only systems and [35] for an implementation in reaction-diffusion systems. Depending on the size of the

molecule populations within a given subvolume and the likelihood of reaction events, the number of events in one step could be found as a Poisson random variable, a Gaussian random variable (i.e., the Langevin method), or a deterministic value (i.e., a continuum model).

2.1.4. Continuum Models

In the limit of large molecule populations, mesoscopic models become *continuum* models, where local molecule concentrations have continuous values and the system is in effect evolving via the solution of a set of partial differential equations with finite element analysis. Such a system is no longer stochastic but has deterministic behavior. COMSOL Multiphysics (see [36]) and ANSYS (see [37]) are commercially-available continuum solvers.

2.1.5. Hybrid Models

There have been multiple attempts to integrate multiple simulation models into a single reaction-diffusion solver. Doing such an integration improves flexibility in the trade-off between accuracy and computational complexity by using a more accurate model only where (or when) it is needed. For example, approaches have been proposed that combine microscopic and mesoscopic models, as in [38, 39, 40]. The hybrid model in [40] has been implemented in the (primarily mesoscopic) solver URDME (see [32]) and the model in [39] was recently implemented in the (primarily microscopic) solver Smoldyn (see [41]). LAMMPS has also coupled its molecular dynamics model to a continuum model; see [42]. Virtual Cell is a continuum solver that added Smoldyn for microscopic behavior; see [43].

2.2. Diffusive Molecular Communication Simulators

The existing generic reaction-diffusion solvers described in Section 2.1 are, in general, useful “sandbox” tools for exploring the dynamics of molecular behavior. However, there are several characteristics that limit their applicability to the study of molecular communication networks. We identify these characteristics as follows:

- Existing generic solvers are not designed to accommodate the behavior of a transmitter that is releasing molecules according to the modulation of a data sequence. It can be possible to release finite pulses of molecules at specific times, for example via commands in Smoldyn, but other arbitrary molecule signal waveforms are not as easily accommodated (e.g., frequency shift keying). Furthermore, the molecule release times would need to be hard-coded into the configuration. Such an approach would be inconvenient for measuring the average bit error probability at a receiver over a large number of randomly-generated transmitter bit sequences, since every bit sequence would need its own corresponding configuration.

- In communications analysis, we are not necessarily interested in accurately observing system behavior *everywhere*. Instead, we are ultimately interested in the behavior at the *receiver(s)* of information. Selecting a solver model typically imposes a particular range of accuracy over the entire simulation environment. Thus, computational resources might be “wasted” to maintain accuracy in regions of the environment that have minimal impact on the receiver(s) of interest. However, this limitation can be mitigated by choosing an appropriate hybrid simulation model.
- Existing generic solvers are not designed for generating channel statistics over a large number of independent realizations. The reliability of a receiver is often measured as the probability that it correctly detects a message sent from a transmitter, which implies that a particular observation (or series of observations) was made by the receiver. If we want to simulate the time-varying probability distributions of receiver observations, i.e., the non-stationary channel statistics, then we may need to simulate one scenario many thousands of times and aggregate the results. Such a functionality is not native to existing generic solvers.

Given the aforementioned limitations in existing generic reaction-diffusion solvers, communications researchers have developed a number of simulators specifically for simulating diffusive molecular communication systems. A recent discussion comparing most molecular communication simulators is in [6]. Here, we highlight the features of simulators whose source code or executables are publicly available. Thus, we exclude simulators that have been described but not released, including NanoNS (Nanoscale Network Simulator) in [44], dMCS (Distributed Molecular Communication Simulator) in [45], and other tools developed internally by authors for their own research.

BNSim (Bacterial Network Simulator) was presented in [16] and is a multi-scale mesoscopic reaction-diffusion solver for simulating interactions between mobile bacteria populations. The bacteria undergo chemotaxis (i.e., run-and-tumble motion) and collisions between them are accounted for. Key chemical reactions are simulated with the complete Stochastic Simulation Algorithm (SSA; see [46]) whereas other reactions are simulated over larger time scales via tau-leaping or the Langevin method; see a review of this and other methods in [47].

BiNS2 (Biological Nanoscale Simulator) was described in [17] and is a microscopic reaction-diffusion simulator for flowing cylindrical environments (e.g., blood vessels). Both transmitters and receivers can be mobile and molecules are detected via receptor reactions. The simulation environment is customizable, can include local obstacles, and can be visualized at run time.

N3Sim was presented in [48, 18] and is a microscopic simulator for a square or unbounded 3D environment. In

addition to Brownian motion, it can account for molecule inertia and collisions between solvent molecules. Circular or spherical emitters can release molecules according to predefined waveforms or a user-defined pattern. Receivers can be squares or circles in 2D and spheres in 3D, and can be either passive or fully absorbing. Notably, N3Sim is the only publicly-available molecular communication simulator (excluding AcCoRD) with a detailed user guide and instructions for use (see [49]).

MUCIN (MolecUlar CommunicatIoN) was presented in [19] and is a microscopic simulator for unbounded 3D environments developed in MATLAB. It models a point or spherical transmitter that releases molecules according to one of several modulation schemes. The spherical receiver can be passive, partially absorbing (i.e., molecules are reflected with some probability), or fully absorbing.

The IEEE P1906.1/Draft 1.0 Recommended Practice for Nanoscale and Molecular Communication Framework, summarized in [50], uses a simulation tool implemented on top of ns-3 as a reference simulator. It is used to compare molecular and electromagnetic communication schemes. However, this simulator uses analytical results from a specific physical environment and does not simulate the environment directly.

Finally, nanoNS3 was introduced in [20] and is a continuum simulator for bacteria-based molecular communication. It is implemented on top of ns-3 and has models implemented for the signal observed at a receiver bacterium, microfluidic channel loss, ON/OFF transmitter modulation, and addressing via signal amplitudes.

2.3. Comparing with AcCoRD

Some of the existing simulators for diffusive MC, namely BNSim and BiNS2, are very detailed for their intended environments and how they can be configured. However, these tools and the other MC simulators were not designed as generic reaction-diffusion solvers, so they are not as flexible as AcCoRD for studying new and different environments. Instead of making exhaustive comparisons between AcCoRD and all aforementioned simulation tools listed in this section, we make a simplified comparison in Table 1. The first “model” listed for each simulator is its primary model, but we emphasize that only AcCoRD was initially designed as a hybrid of simulation models. Furthermore, AcCoRD is the only generic reaction-diffusion solver (i.e., “sandbox” simulator) that can accommodate molecule sources that modulate a sequence of randomly-generated data. We believe that this combination facilitates studying the performance of a variety of diffusion-based molecular communication systems.

We will find it most insightful to make a direct comparison between AcCoRD and the microscopic reaction-diffusion solver Smoldyn. AcCoRD and Smoldyn have a number of similar underlying algorithms and features, such as for chemical reactions and hybrid microscopic-mesoscopic interfaces, which will be discussed in greater

Table 1: Simplified comparison of simulation tools with AcCoRD. Under “Model”, “MD” refers to molecular dynamics, “Micro” refers to microscopic, “Meso” refers to mesoscopic, and “Cont” refers to continuum. Simulators that use more than one model list the primary model first. For each simulator, we also indicate whether each is a generic reaction-diffusion solver (i.e., “sandbox”) and whether they can release molecules according to the modulation of a data sequence. The IEEE P1906.1 reference simulator described in [50] is omitted because it does not simulate the environment directly.

Simulator	Model	Sandbox	Modulate Data
LAMMPS [27]	MD/Cont	Yes	No
Smoldyn [15]	Micro/Meso	Yes	No
URDME [32]	Meso/Micro	Yes	No
COMSOL [36]	Cont	Yes	No
ANSYS [37]	Cont	Yes	No
Virtual Cell [43]	Cont/Micro	Yes	No
BNSim [16]	Meso	No	No
BiNS2 [17]	Micro	No	Yes
N3Sim [18]	Micro	No	Yes
MUCIN [19]	Micro	No	Yes
nanoNS3 [20]	Cont	No	Yes
AcCoRD	Micro/Meso	Yes	Yes

detail in Section 4, Appendix B, and Appendix C. Smoldyn, which has been in active development for over a decade, has a more mature code base, more features such as more options for defining physical environments and surface interactions, and the flexibility to change parameters while a simulation is in progress. However, we identify the following primary advantages for AcCoRD:

- AcCoRD addresses the aforementioned limitations of generic reaction-diffusion solvers to facilitate adoption for communications analysis. In particular, AcCoRD can accommodate the release of molecules according to the modulation of a random binary sequence. Also, simulation realizations can be easily aggregated (even if they were run on different computers) to compile the simulation statistics.
- AcCoRD implements some microscopic behavior in continuous time, whereas microscopic behavior in Smoldyn is completely discrete. Therefore, the accuracy of some phenomena in AcCoRD (namely, zeroth order and some first order reactions) is independent of the chosen time step.
- A primary motivation for AcCoRD was the integration of microscopic and mesoscopic models to enable flexibility in local accuracy, as we initially proposed in [22]. Smoldyn is primarily a microscopic simulator that recently added a mesoscopic “module” that is not as well documented as the rest of the platform. Thus, even though AcCoRD and Smoldyn both implement transitions between the two models using

the hybrid model in [39], the integration in AcCoRD is central to the simulator’s design. In AcCoRD, it is relatively seamless for a user to apply both models. Furthermore, we accommodate adjacent mesoscopic regions that have subvolumes of different sizes, whereas Smoldyn does not.

- AcCoRD includes visualization tools that were developed in MATLAB, which make them accessible and familiar to a wide research audience. Smoldyn can display animations of simulation progress and save images for future use, but the images must be displayed and captured *online*; i.e., while the environment is being simulated. AcCoRD performs its visualization *offline*; a user can preview the physical environment without running a simulation, and simulation output can be processed to either generate images or make a video with the freedom to choose what environment features to display and at what times.
- Smoldyn uses the Mersenne Twister (see [51]) as its random number generator (RNG). The Mersenne Twister is a common RNG and the default RNG in MATLAB. However, the permuted congruential generator (PCG) family of RNGs, recently proposed in [52], is claimed to have improved statistical quality. A detailed discussion of the PCG’s advantages is outside the scope of this work, but we note that it has a faster generation speed and a smaller code footprint. AcCoRD uses a PCG RNG that we have confirmed in internal testing to be faster than an efficient implementation of the Mersenne Twister.

3. System Model Components

We have introduced AcCoRD as a “sandbox” reaction-diffusion solver for molecular communications design and analysis. Therefore, before we discuss the underlying theory (in Section 4) or the implementation of AcCoRD’s algorithms (in Section 5 and Appendix C), we describe the system model components that are available to the user. In this section, we present these components, which are categorized into *regions*, *actors*, *chemical parameters*, and *global settings*. We also summarize how these components are included in an AcCoRD configuration file. A sample environment demonstrating many of these components is shown in Fig. 5 (which is also shown in Video 1 in [23]). Throughout this section, most parameters that can be defined by the configuration are *italicized*. A listing of constraints and limitations on model components are included in Appendix A.

3.1. Regions

Regions are the literal “building blocks” of a simulation environment. They define the physical space where molecules can move or be created. Other components, i.e.,

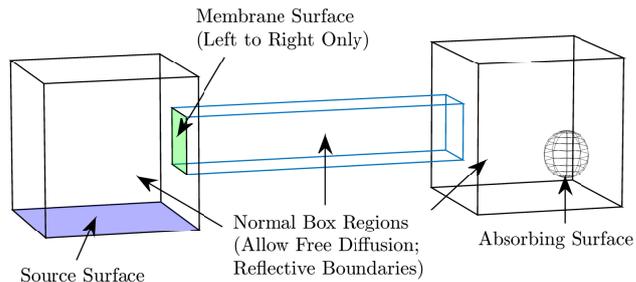


Figure 5: Example of a complex simulation environment with many of the system model components described in Section 3. There are normal and surface regions, actors, and multiple chemical reactions. The entire system is microscopic. Two normal cubic regions are joined by a normal rod region with a membrane surface region at one end (green), such that the membrane’s neighbors are the left cube and the rod, and the rod’s neighbors are the membrane and the right cube. The membrane enables molecules to pass from left to right only. The left cube has a neighboring surface region (blue) that is a source of molecules. The only *parent/child* region relationship is between the right cube and the absorbing spherical surface inside of it. Actors observe the entire environment so that molecule coordinates are generated for the sample video, which is available as Video 1 in [23].

actors and chemical reactions, can be defined for all or some subset of an environment’s regions. As long as some (minor) constraints are satisfied, regions can be placed in solitude, adjacent to other regions, or nested inside of other regions. A *parent* region is one that has another region (i.e., its *child*) nested inside of it. Generally, if a region’s face is adjacent to another region, then those regions are automatically classified as neighbors. As long as neighboring regions are not separated by a surface region that restricts molecule transitions, then molecules can move freely between them.

There are three ways to classify a region:

1. *Shape*. AcCoRD is primarily 3D and the implemented 3D shapes are spheres and rectangular boxes. 2D rectangles are also implemented.
2. *Type*. A region can be *normal* or a *surface*. Normal regions occupy all of the space that they are defined in, except where they are a *parent* to other regions nested inside. A 3D surface region can have either a 3D shape, such as a box or sphere, or it can also have a 2D shape (i.e., a rectangle). 2D rectangle regions are implemented but have only been tested as surfaces to normal 3D regions. Surfaces also have an associated “direction” as defined by the *surface type*. A one-sided surface means that molecules can only approach and interact with one of the two sides and not both. A two-sided surface, defined as a *membrane*, can have molecules interact from either side. However, all surfaces are reflecting by default, un-

less the user defines chemical reactions at a surface to enable other interactions (as described later in this section).

3. *Model*. Each region uses either a *microscopic* or *mesoscopic* model. All molecules in microscopic regions are tracked individually and evolve according to a global *microscopic time step*. Mesoscopic regions keep track of how many of every type of molecule are in each subvolume. In a given simulation, we define the union of all microscopic regions as the *microscopic regime*, and the union of all mesoscopic regions as the *mesoscopic regime*. All regions, whether they are microscopic or mesoscopic, are partitioned into *subvolumes*. Subvolumes are the foundation of a mesoscopic model, but they are also useful for the implementation of microscopic regions when determining where regions are located relative to each other. Spherical regions are always microscopic and have only one subvolume.

The location of a region is defined by its *anchor coordinate*, which for example is the center of a sphere or the “lower corner” $\{x, y, z\}$ coordinate of a box. The size of a sphere is defined by its *radius*, which can in principle be any positive real number (including ∞ for an unbounded environment). The sizes of boxes are constrained by a global *subvolume base size* h_{base} . The length of a box along each dimension $\{x, y, z\}$ is defined as an integer multiple of h_{base} , and each subvolume in a box is a cube whose length, the *region subvolume length* h , is also an integer multiple of h_{base} . We discuss some of the physical and computational considerations for choosing an appropriate subvolume length in Section 5.2. 2D rectangle regions have a length of 0 along one of the three dimensions. Defining h_{base} facilitates the placement of regions (and also actors) by avoiding issues due to floating point operations (e.g., when $2 \times 1 \neq 2$ due to different data types or floating point rounding errors).

Given the above definitions, regions can be placed almost arbitrarily. Generally, regions can be placed adjacent to each other or inside of each other (i.e., nested), such that the *parent* outer region is identified by the *child* inner region. Any two regions that are adjacent or has one nested directly inside the other are detected as *neighbors*. Surface regions are usually only defined along the outer boundaries of normal regions, unless the surface is a 3D shape nested inside of a normal 3D region. For clarity of exposition, we list more details of the constraints on region placement in Appendix A.

There are two additional properties that are not specific to any one region but are needed when a simulation has both a microscopic regime and a mesoscopic regime. These properties control how molecules can transition from one regime to the other (i.e., at the hybrid interface), and are described in further detail in Section 4.2.2. The first property selects whether the mesoscopic subvolumes at the hybrid interface are assumed to be small rela-

tive to the average displacement in the microscopic regime. The second property defines how far away a microscopic molecule should be from the boundary, either before or after it diffuses, to ignore the possibility that it entered the mesoscopic regime *during* the diffusion step.

3.2. Actors

Actors in AcCoRD provide the interface to a simulation by enabling input of molecules or observing molecules as output. The two classes of actors are *active* and *passive*. These classes share some common properties but otherwise have unique behavior. Active actors (e.g., transmitters) enable input by creating molecules according to the modulation of a binary data sequence. Passive actors (e.g., receivers) enable output by observing the number of molecules that are present at some location. Both classes of actors are currently immobile and *independent*, i.e., their behavior is established by the user and does not depend on that of the system or other actors. However, the generic actor design facilitates the future implementation of *dependent* actors which will have coupled behavior. We discuss this further in Section 8.

All actors can be placed using one of two methods. In the first method, an actor is defined as the union of a subset of the regions. This method is preferred when an actor is intended to exist over an entire region or regions, and enables an actor to be disjoint or to have non-standard shapes (e.g., if an actor is defined either over disjoint regions or over some region and not that region’s nested children). In the second method, an actor is defined as a single virtual shape. Currently, all region shapes are valid actor shapes, and it is also possible to define an active actor as a *point*. Actors defined with the second method are always defined over the entire specified shape (i.e., and not just its surface).

It is important to clarify that actors are virtual and *do not* have a tangible physical presence in a simulation environment. They only define where molecules can be added or observed. Regions must be used to impose physical boundaries. For example, if a user wants a receiver that is an absorbing sphere, then the configuration needs to define a spherical surface region, the absorbing chemical reaction at the surface (see the following subsection), and a passive actor to observe the molecules that are absorbed. However, if the user wants a receiver that is a passive sphere within some larger environment, then it is most computationally efficient to just define a spherical actor without an accompanying region (due to the significant overhead to identify transitions between microscopic regions).

Every actor has its own *start time* to begin its first action (i.e., releasing or observing molecules). The actions are repeated according to an *action interval*, until either a *maximum number of actions* has been reached or the simulation time is complete.

The release pattern of an active actor is based on its binary *bit sequence*, which can be defined by the user or

randomly generated according to the specified independent probability of bit-1. Multiple bits can be used in each action interval (i.e., symbol interval). Currently, *concentration shift keying* (CSK) is implemented, where one type of molecule is released. Other types of modulation can be added in future development; see [53] for examples. Molecules for each symbol are released over a *release interval*, which can be as short as 0 seconds or as long as desired (even longer than the action interval). An actor can be configured to behave as a noise source and release molecules continuously by setting the probability of bit-1 to 1 and the release interval to be equal to the action interval. The precise number of molecules created and their release times within the release interval can be random (i.e., generated via a Poisson arrival process, such that the time between consecutive molecule releases is an exponential random variable) or deterministic (by dividing the release interval into discrete *slots* and releasing molecules at the start of each slot). Every added molecule is placed at a uniformly-distributed random location within the actor. If an active actor is being recorded, then its bit sequence is written to the simulation output file.

The behavior of passive actors is simpler. Each passive actor acts by counting the number of molecules of the *specified types* that are within their defined space. It is also possible to *record molecule positions*, which are copied for molecules in the microscopic regime and randomly generated within the corresponding subvolume in the mesoscopic regime. Another option is to *save the environment time* associated with each observation.

3.3. Chemical Properties

The chemical properties describe the molecules that can exist in the environment and how they can interact via reactions. Even if there are no chemical reactions, the number of molecule types and their *diffusion coefficients* must be defined. There are two methods to represent different states of the same molecule species, for example to specify that a molecule should not diffuse once it has been absorbed by a surface. In the first method, the user can define two types of molecules, set one of them to have a diffusion coefficient of zero, and make that molecule the product of the other molecule’s absorption reaction. In the second method, the user can define one type of molecule and set that molecule’s local diffusion coefficient to zero at the surface region (via an optional region property).

Chemical reactions define processes for molecules to be created, destroyed, or transformed; refer to [11, Ch. 9] for an elementary discussion. Every chemical reaction has a set of *reactants* (which the reaction consumes), a set of *products* (which the reaction creates), and some way to indicate the likelihood of the reaction occurring (typically a *reaction rate*). A molecule that is a reaction catalyst and facilitates the reaction without being consumed (i.e., an enzyme) can be represented as both a reactant and a product. Reactions with no reactants, one reactant, and two reactants are referred to as *zeroth order*, *first order*,

and *second order*, respectively. Reactions with more reactants are typically decomposed into a sequence of elementary reactions that are zeroth, first, or second order. First and second order reactions are also known as unimolecular and bimolecular reactions, respectively.

We classify each chemical reaction in AcCoRD as either a *surface reaction* or a *non-surface reaction*. Surface reactions can be either 1) an interaction between one molecule and a surface *region*, e.g., *absorption* (i.e., consumption) by a surface, *desorption* (i.e., release) from a surface, and transition across a *membrane* surface, or simply 2) a reaction where the reactants *must* be on the surface. We consider adsorption (i.e., sticking to a surface) to be reversible absorption. Non-surface reactions do not require a surface region, but can also occur at a surface. Ligand-receptor surface binding is *not* considered a surface reaction under our definition, because one of the reactants (i.e., the ligand) is initially not on the surface and it reacts with the receptor and *not* the surface³. Every chemical reaction is associated with a default list of regions where it can occur, and this can be set as *everywhere* (i.e., in all surface regions for a surface reaction or in all normal regions for a non-surface reaction) or nowhere. Regions that are exceptions to the default placement can also be listed.

Reaction rates for zeroth, first, and second order reactions have units $\text{mol} \cdot \text{s}^{-1} \text{m}^{-3}$, s^{-1} , and $\text{mol}^{-1} \text{s}^{-1} \text{m}^3$, respectively, where “mol” is number of molecules. Second order reactions that can occur in the microscopic regime also need a *binding radius* r_B which is used instead of the reaction rate (in general, the binding radius can be derived from the reaction rate, but this is not currently part of AcCoRD’s implementation; see [15] for further details). Bimolecular reactions in the microscopic regime with more than one product also have an *unbinding radius* r_U to define how far apart products should be placed. We note that the random generation of molecules by an active actor is also modeled as a zeroth order reaction but is distinct from other zeroth order reactions because the generation rate for the actor is modulating a data sequence.

The named surface interaction reactions, i.e., absorption, desorption, and membrane reactions, are assumed to be first order (since they have one molecule interacting with a surface) and have additional properties. These reactions can be classified as *reversible* if they define a corresponding *reverse reaction*. They also need a *surface transition probability*, which determines the calculation in [24] (and summarized in Appendix B) that we use to calculate the probability of the reaction occurring. The three options for this probability are as follows:

³Our classification of reactions is similar but not identical to classifying reactions as heterogeneous or homogeneous; see [54, Ch. 15]. Heterogeneous reactions have reactants, products, or catalysts in different states (e.g., liquid and solid), whereas homogeneous reactions occur entirely in a single state. Surface reactions in AcCoRD are heterogeneous when a molecule interacts with a surface region, but non-surface reactions can also be heterogeneous, such as in the case of ligand-receptor surface binding.

1. “*Normal*”: The reaction rate is treated as if the reaction is a *non-surface* first order reaction. This option can accommodate “perfect” absorption (i.e., when the absorption rate is infinite), but otherwise its application is limited.
2. “*Mixed*”: We assume that the distribution of molecules near the surface is in the well-mixed state, such that the concentration is uniform. This option is fast to calculate and accurate for systems that are actually well-mixed.
3. “*Steady state*”: We assume that the distribution of molecules near the surface is in the steady state, such that the concentration is constant but can vary locally. This option was shown in [24] to have excellent accuracy, even when the system is transient. Also from [24], the reverse reaction rate is accounted for when determining this reaction probability.

Membrane reactions should not have any product molecules defined (since we assume that the product molecule is the same type as the reactant) and they need to be classified as *inner* or *outer* to indicate the side of the membrane that the reactant can approach from. Non-membrane surface reactions need to explicitly indicate whether each product is automatically detached from the surface (although this is generally intended only for desorption reactions) and if so then how the products are placed. The options for placing detached product molecules are as follows:

1. “*Leave*”: Molecules are placed in the normal region at the point where they were bound to the surface. This option is the simplest but is only accurate if the molecule detaches at the end of the microscopic time step.
2. “*Full diffusion*”: Molecules diffuse perpendicular to the surface for the elapsed time since the detachment occurred. This option should be the most accurate in cases of irreversible desorption.
3. “*Steady state diffusion*”: Molecules diffuse perpendicular to the surface for an unknown period of time that is no greater than the microscopic time step. This option is the most appropriate when the detachment reaction has a reverse adsorption reaction whose surface transition probability is steady state, because then the precise adsorption time is also unknown. The placement distribution for this method was derived in [24] in consideration of the steady state transition probability.

The methods for placing molecules according to the “full diffusion” and “steady state diffusion” options are discussed in Appendix B.2.

3.4. Other Settings

Every simulation starts at time $t = 0$ but needs a specified *final time* and *number of times to be repeated*. The *random number seed* is used to initialize the PCG RNG

(see [52]), although it can be over-ridden with a different seed via an extra command line input argument. The *prefix of the simulation output* file is specified such that the random number seed (that is actually used) is appended to it. Finally, the user can also choose to *override the display of warnings* from the loading of the configuration file and control the *maximum number of progress updates* that will be printed to the console screen with an estimate of the remaining run time to complete the simulation. There will be no more than one progress update per independent realization, since it is difficult to estimate the time remaining while within a realization.

3.5. Configuration Files

AcCoRD’s configuration files are written in JavaScript Object Notation (JSON) format. JSON is a lightweight data interchange format, easy to read, and has parsers available in many programming languages; see [55]. AcCoRD’s configuration files accommodate all of the system model components introduced in this section with the following consistent format:

- Output Filename
- Switch to over-ride configuration warnings
- Simulation Control
 - Number of repeats and random number seed ψ
 - Final simulation time t_{end} and microscopic time step Δt
 - Hybrid interface parameters
- Chemical Properties
 - Number of types of molecules and their diffusion coefficients D_m
 - One object describing each chemical reaction
- Environment
 - Subvolume Base Size h_{base}
 - Region Specification
 - * One object describing each region
 - Actor Specification
 - * One object describing each actor

As long as the correct nesting of parameters is maintained, JSON files can be arbitrarily reordered. Even though JSON does not support comments, extra fields can be created to include additional information. Many of the sample configuration files included with the AcCoRD source code have “Notes” fields for comments.

4. Reaction-Diffusion Theory and Algorithms

In this section, we describe the underlying theory for the microscopic and mesoscopic simulation models. We present the equations and algorithms for modeling diffusion, chemical reactions in solution, and surface reactions. This section includes our derivation of the transition rate between mesoscopic subvolumes of different sizes, and the continuous reaction event time for first order chemical reactions that are known to have occurred in the current microscopic time step.

Before we describe the analytical details of the individual phenomena, we establish some global model definitions and notation. The total simulation environment is partitioned into a set of regions Ω_R . Each region is either in the microscopic regime Ω_{micro} or the mesoscopic regime Ω_{meso} , and $\Omega_{\text{neigh},r}$ is the set of regions that are neighbors of region r . The mesoscopic regime is partitioned into the set of subvolumes Ω_{sub} . The set of defined molecule types is Ω_{Mol} and the set of chemical reactions is Ω_C . The number of molecules of the m th type that are in the s th mesoscopic subvolume is $U_{s,m}$.

4.1. Model Evolution

The most intuitive way to distinguish between the two simulation models is to summarize how they behave over time and space. The microscopic model describes molecule locations over continuous space but evolves in discrete time steps of constant size Δt . For each time step, *every* microscopic molecule can diffuse and react.

The mesoscopic model describes molecule locations over a discrete grid (of subvolumes) but evolves over continuous time. In its most accurate form (i.e., without tau-leaping, which we will consider in future development), only *one* mesoscopic event occurs at a time. The potential events are every possible chemical reaction in each subvolume and every possible transition via diffusion between adjacent subvolumes. Each potential event is assigned a *propensity*, α , such that the probability of the event occurring within the infinitesimal time step δt is $\alpha \delta t$.

The integration of the two models within the overall AcCoRD simulation algorithm is described in Section 5, along with a discussion of their computational complexity and comments on the appropriate selection of time step Δt and the mesoscopic subvolume size. The implementation of both individual models is discussed in greater detail in Appendix C.

4.2. Diffusion

Diffusion is modeled in both regimes with diffusion coefficients, which describe the variance of motion of individual molecules by assuming that the molecules are moving in a continuum of solvent (e.g., air, water, or blood). The diffusion coefficient of the m th molecule type is D_m , which we will occasionally write as D when referring to an arbitrary molecule.

4.2.1. Diffusion in One Regime

In the microscopic regime, every molecule tries to diffuse in every time step. In the absence of collisions with surfaces or other molecules, the displacement of a single molecule in one time step is $n\sqrt{2D_m\Delta t}$ along each dimension, where n is a normal random variable with mean 0 and variance 1. n is independently drawn in each dimension for every molecule.

In the mesoscopic regime, it is assumed that all molecules within a given subvolume are uniformly distributed. The propensity that some molecule diffuses from a subvolume into a neighboring subvolume is directly proportional to the number of molecules in that subvolume that are of the same type. For a uniform grid of square or cubic subvolumes of length h , where neighboring subvolumes share an *entire* face, the diffusion propensity for the m th molecule type from the s th subvolume to the q th subvolume is given as [39, Eq. (1.6)]

$$\alpha_{s,q,m} = \frac{D_m}{h^2} U_{s,m}, \quad (1)$$

assuming that subvolumes s and q share a face (otherwise, $\alpha_{s,q,m} = 0$).

In AcCoRD, mesoscopic subvolumes can have different sizes and so they might have faces that are only partially shared. Thus, we need a more general expression to describe the diffusion propensity. As we considered for the 2D case in [22], we start with the propensity from a 1D subvolume of length h_s to one of a different length h_q . This was previously derived as [56, Eq. (15)]

$$\alpha_{s,q,m} = \frac{2D_m}{h_s(h_s + h_q)} U_{s,m}. \quad (2)$$

The advantage of (2) over (1) is that (2) accounts for the size of the destination subvolume while satisfying Fick's law for the diffusion flux. The impact is that a molecule will be less likely to enter a larger neighbor than one that is the same size. While this might be an un-intuitive result, it can be shown that (2) will maintain a uniform distribution of molecules between the two subvolumes whereas (1) will not.

Eq. (2) immediately applies to the 3D case *if* the face shared by subvolume s with subvolume q is completely covered by subvolume q . However, this does not always occur in the 3D case. In [22], we scaled (2) by the *relative overlap length* for the 2D case, i.e., the fraction of the line segment that subvolume s shares with subvolume q . Here, we consider the *overlap area*, A_o , which is the size of the shared surface between subvolume s and subvolume q (see Fig. 6). The likelihood of a molecule diffusing from subvolume s to subvolume q should be scaled by A_o/h_s^2 , which is the fraction of the face of subvolume s that is actually shared with subvolume q . Thus, we can write the propensity for the m th molecule type to diffuse from subvolume s to subvolume q as

$$\alpha_{s,q,m} = \frac{2D_m A_o}{h_s^3(h_s + h_q)} U_{s,m}. \quad (3)$$

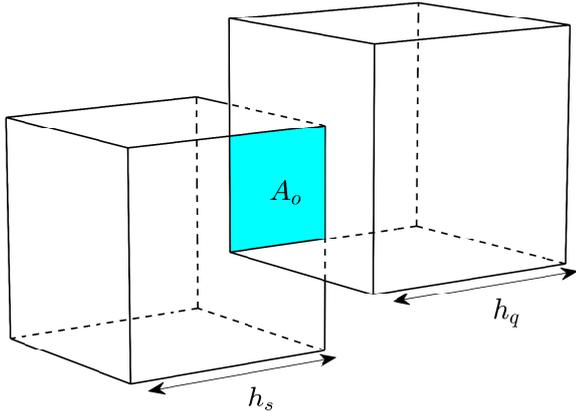


Figure 6: Two 3D mesoscopic subvolumes that partially overlap. The overlap area with size A_o is shown in aqua blue. Such overlap areas can occur in order to accommodate regions that have subvolumes of different sizes or complex environment geometries. In this example, the two subvolumes actually have equal widths, i.e., $h_s = h_q$.

where $A_o \leq \min\{h_s^2, h_q^2\}$.

4.2.2. Hybrid Diffusion Between Microscopic and Mesoscopic Regimes

When both regimes exist, then we have hybrid diffusion and we need to account for diffusion across the interface between the two regimes. We adopt the transition rules described in [39]. The rules define how an individual molecule in the microscopic regime can be placed in a mesoscopic subvolume, and how a diffusion event in a mesoscopic subvolume can result in a new microscopic molecule. We summarize both processes here.

A molecule in the microscopic regime has two ways to enter the mesoscopic regime. First, a molecule is automatically placed in a mesoscopic subvolume if its diffusion trajectory crosses the subvolume. Otherwise, if the molecule's destination region (i.e., after diffusion) has mesoscopic neighbors, then we still consider the possibility that the molecule entered the mesoscopic regime *during* the time step (since actual diffusion is not along a straight line). The probability that the molecule entered mesoscopic region r *within* arbitrary time step δt , P_{intra} , is given by [39, Eq. (1.9)]

$$P_{\text{intra}} = \exp\left(-\frac{l_{r,i}l_{r,f}}{D_m\delta t}\right), \quad (4)$$

where $\{l_{r,i}, l_{r,f}\}$ are the closest distances from the molecule to region r at the start and end of the diffusion step, respectively. We introduce a maximum distance l_{max} to control computational requirements, i.e., (4) is ignored if either of $\{l_{r,i}, l_{r,f}\}$ are greater than l_{max} . Alternatively, defining relatively small microscopic regions that neighbor the mesoscopic regime will also limit the number of checks using (4), although this will also increase the number of checks for diffusion across microscopic region boundaries.

For molecules that originate in the mesoscopic regime, we need to consider the propensity for a molecule to enter the microscopic regime and where to place the molecule when a diffusion event occurs. For ease of implementation, we assume that if a mesoscopic subvolume is adjacent to microscopic region r along some face, then the *entire* face is shared with region r , i.e., the overlap area for subvolume s is $A_o = h_s^2$. Thus, the propensity for the m th molecule type to diffuse from subvolume s to region r is [39, Eqs. (1.10), (1.11)]

$$\alpha_{s,r,m} = \frac{2}{h_s} \sqrt{\frac{D_m}{\pi\Delta t}} U_{s,m}, \quad (5)$$

which replaces (3) when the neighboring subvolume is in a microscopic region. We note that (5) depends on the microscopic time step Δt , so a smaller time step will result in a higher propensity to leave subvolume s .

If a diffusion event defined by the propensity in (5) occurs, then a microscopic molecule must be created. Specifically, tangential and normal coordinates must be chosen, since the molecule is able to diffuse away from the mesoscopic surface until the next microscopic time step. Without loss of generality in presentation (since any other orientation is analogous), we assume that the diffusion event occurs at a subvolume s whose shared face with the microscopic region is in the yz -plane and centered at the origin with the microscopic region in positive x , as shown in Fig. 7. Then, the probability distribution to initialize the microscopic molecule in the x -direction is [39, Eq. (1.12)]

$$f_x(x) = \sqrt{\frac{\pi}{4D_m\Delta t}} \operatorname{erfc}\left(\frac{x}{\sqrt{4D_m\Delta t}}\right), \quad x > 0, \quad (6)$$

where $\operatorname{erfc}(\xi) = 1 - \operatorname{erf}(\xi)$ is the complementary error function (from [57, Eq. (3.1.2)]), and the error function is [57, Eq. (3.1.1)]

$$\operatorname{erf}(\xi) = \frac{2}{\sqrt{\pi}} \int_0^\xi \exp(-\gamma^2) d\gamma. \quad (7)$$

As presented in [24, Eq. (35)], an efficient approximation to sample from the distribution in (6) is to generate a uniform random number u between 0 and 1 and then x is

$$x = \frac{0.729614u - 0.70252u^2}{1 - 1.47494u + 0.484371u^2} \sqrt{2D_m\Delta t}. \quad (8)$$

For the tangential directions $\{y, z\}$ there are two models presented in [39]. Both models make assumptions about the relative size of the mesoscopic subvolumes in order to simplify the infinite sum that appears when balancing the diffusion distributions at the interface. The models either assume that the mesoscopic subvolumes are relatively *small* (i.e., that the spatial resolution of the mesoscopic regime is comparable to that of the microscopic regime) or that the subvolumes are relatively *large* (i.e., that the spatial resolution of the mesoscopic regime

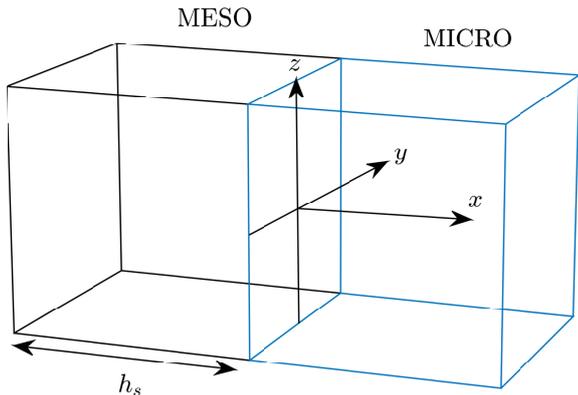


Figure 7: View of hybrid interface between a mesoscopic subvolume (“MESO”) and a microscopic region (“MICRO”). Coordinates are consistent with the microscopic molecule placement distributions in (6), (9), and (10).

is larger than that of the microscopic regime). In the first model, where the mesoscopic subvolume is assumed to be relatively *small*, the model assumes that $h_s \sim \sqrt{D_m \Delta t}$ as $h_s \rightarrow 0$ and $\Delta t \rightarrow 0$. This assumption, which is adopted by the hybrid implementation in Smoldyn (see [41]), uses a triangular distribution to place the molecule along each tangential dimension, i.e., [39, Eq. (2.14)]

$$f_y(y) = \frac{1}{h_s} \left(1 - \frac{|y|}{h_s} \right), \quad -h_s < y < h_s, \quad (9)$$

for y (and analogously for z). In the second model, where the subvolume is assumed to be relatively *large*, the model assumes that $h_s \gg \sqrt{D_m \Delta t}$. This results in a uniform distribution along each tangential dimension, i.e., [39, Eq. (2.16)]

$$f_y(y) = \frac{1}{h_s}, \quad -\frac{h_s}{2} < y < \frac{h_s}{2}, \quad (10)$$

for y (and analogously for z). We implement *both* the large and small hybrid subvolume models in AcCoRD, so the user can choose the most appropriate model in consideration of the relative size of the subvolumes at the hybrid interface.

4.3. Chemical Reaction Kinetics

The accurate modeling of a chemical reaction relies on knowledge of the reaction’s *elementary* steps, which describe the actual reaction steps that occur at the molecular level; see [11, Ch. 9]. If a reaction’s elementary steps are known, then the reaction kinetics can be determined and simulation is possible. The three common classes of elementary reactions are zeroth order, first order, and second order (higher order elementary reactions, which rely on the unlikely simultaneous collision of three or more molecules, are very rare). Here, we describe each of the three common orders and how they are simulated. We also discuss the special cases of surface reactions.

4.3.1. Zeroth Order Reactions

Zeroth order reactions are of the form



such that product molecules are spontaneously created in the propagation environment. The zeroth order reaction rate constant, k_0 , with units $\text{mol} \cdot \text{s}^{-1} \text{m}^{-3}$, specifies the number of molecules created per second per cubic meter of solvent. When a zeroth order reaction is defined on a 2D surface, we define the reaction rate with units $\text{mol} \cdot \text{s}^{-1} \text{m}^{-2}$. We note that a true zeroth order chemical reaction is unphysical, but it models a source that adds molecules to the environment at some rate. In fact, we implement the addition of molecules by active actors as zeroth order reactions with a time-varying rate that is modulated by its data sequence (unless the actor is configured to release its molecules instantaneously).

A zeroth order reaction is a Poisson process, i.e., the times between reaction events are continuous, independent, and have a constant rate; see [58, Ch. 5]. Realizations of the time t_c between consecutive events of zeroth order reaction c with rate $k_0 = k_c$ in region r can be obtained by generating exponential random variables via the inverse transform method, i.e., [58, Ch. 15]

$$t_c = -\frac{\log u}{k_c V_r}, \quad (12)$$

where u is an independent random number uniformly distributed between 0 and 1, V_r is the volume (or area if applicable) of region r , and \log is the natural logarithm. In the microscopic regime, we create molecules on a continuous timescale, so zeroth order reactions are directly simulated using (12). Molecule locations are created uniformly within the region. In the mesoscopic regime, we need to represent the reaction with a corresponding propensity. We will see in Appendix C.4 that *all* events in the mesoscopic regime occur at times according to a sequence of exponential random variables. For zeroth order reaction c , the propensity in subvolume s is [56, Eq. (6)]

$$\alpha_{s,c} = k_c V_s, \quad (13)$$

where V_s is the volume (or area) of subvolume s .

In the case of actor a being active and releasing molecules at random times, we use (12) in both the microscopic and mesoscopic regimes by replacing V_r with the actor volume V_a . This simplifies the implementation by minimizing changes to the mesoscopic algorithm when an actor starts or stops adding molecules. When an active actor creates a molecule that should be placed in the mesoscopic regime, then it is added to the corresponding subvolume.

4.3.2. First Order Reactions

First order (or unimolecular) reactions are of the form



such that an A molecule needs to exist and it is transformed into one or more products at a rate k_1 , which has units s^{-1} . In the mesoscopic regime, the propensity of first order reaction c with rate $k_1 = k_c$ in subvolume s is [56, Eq. (6)]

$$\alpha_{s,c} = k_c U_{s,m}, \quad (15)$$

where the reactant is a molecule of type m . Product molecules are placed in the same subvolume as the reactant.

Our implementation of first order reactions in the microscopic regime is more involved. If a non-surface reaction c is the only first order reaction for which a molecule of type m is the reactant, then the probability of a given molecule of type m reacting within arbitrary time interval δt is [15, Eq. (13)]

$$\Pr(\text{Reaction } c) = 1 - \exp(-k_c \delta t), \quad (16)$$

but if the molecule can participate in multiple non-surface first order reactions, then the probability of reaction c occurring is [15, Eq. (14)]

$$\Pr(\text{Reaction } c) = \frac{k_c}{\sum k_{m,1}} \left[1 - \exp\left(-\delta t \sum k_{m,1}\right) \right], \quad (17)$$

where $\sum k_{m,1}$ is the sum of the rate constants of all non-surface first order reactions for which a molecule of type m can be a reactant. For both (16) and (17), we can simulate whether reaction c occurs by generating a uniform random variable u and comparing it with the corresponding probability.

In AcCoRD, we implement non-surface first order reactions in the microscopic regime on a continuous time scale, which makes the accuracy of these reactions independent of the microscopic time step Δt . So, when first order reaction c occurs *and* at least one product molecule is created, we need to determine the precise reaction time within the current microscopic time step. We consider the time interval $[0, \Delta t]$, i.e., relative to the current microscopic step. Generally, we assume that the candidate reactant molecule was created within this interval at time t_i , i.e., $0 \leq t_i < \Delta t$ (and if the molecule previously existed, then $t_i = 0$). The time remaining for this molecule to react is $\delta t = \Delta t - t_i$. From mesoscopic theory, we can generate the reaction time for a first order reaction as an exponential random variable via

$$t_1 = -\frac{\log u}{\sum k_{m,1}}, \quad (18)$$

but this time is unconstrained. We need to generate a *constrained* exponential random variable so that the reaction time $t_1 \in [0, \delta t]$. We can achieve this by constraining the uniform random variable u . If we set $t_1 = \delta t$ and rearrange (18), then we can determine that the minimum value of u is

$$u_{\min} = \exp\left(-\delta t \sum k_{m,1}\right). \quad (19)$$

Thus, the time of reaction c can be simulated using (18), where u is between u_{\min} and 1. The reaction time of reaction c relative to the current microscopic time step is then $t_c = t_i + t_1$.

The absorption and membrane surface reactions, where a molecule in solution interacts with a surface, are special cases of first order reactions that can only occur *if* the molecule attempts to diffuse across the surface. Furthermore, their corresponding reaction rates are more strictly referred to as reaction coefficients with units m/s . Given that a molecule's diffusion trajectory intersects a reactive surface, then the surface reaction occurs with some probability. If the reaction does not occur, then the molecule is reflected off of the surface. We adopt the surface reaction probabilities derived for Smoldyn in [24], which apply to surfaces in the microscopic regime. For clarity of exposition, the surface reaction probabilities and details for placing product molecules are described in Appendix B. The primary difference between Smoldyn's implementation and AcCoRD's is that surface reactions in AcCoRD occur over the sub-microscopic time interval $\delta t \leq \Delta t$, which is based on a molecule's *creation time*. However, in order to adopt the analysis in [24], we generally assume (unless otherwise noted) that the precise reaction times for surface reactions are *unknown*. Thus, the accuracy of absorption and membrane reactions depends on the microscopic time step.

As noted in [24], it is most accurate to also consider the possibility that a molecule reacts with a surface even if its diffusion trajectory does not intersect the surface. This is analogous to considering that a microscopic molecule might enter and exit the mesoscopic regime *within* the time step, which occurs with probability (4) when the reactive surface is a flat plane. The author of [24] claimed that the corresponding potential for improving accuracy is minimal, so we also ignored this possibility in AcCoRD's implementation. However, given the small time steps that are needed for accurate surface reactions in Section 7, we can re-visit the benefits of this simplification in future work.

4.3.3. Second Order Reactions

Second order (or bimolecular) reactions are of the form



i.e., an A molecule collides with a B molecule and they are transformed into one or more products at a rate k_2 , which has units $\text{mol}^{-1}\text{s}^{-1}\text{m}^3$ (in 3D). In the mesoscopic regime, the propensity of the second order reaction c with rate $k_2 = k_c$ in subvolume s is [56, Eq. (6)]

$$\alpha_{s,c} = \frac{k_c U_{s,A} U_{s,B}}{V_s}, \quad (21)$$

where $U_{s,A}$ is the number of A molecules and $U_{s,B}$ is the number of B molecules in subvolume s , respectively. The product $U_{s,A} \cdot U_{s,B}$ represents the number of potential reactive collisions between specific A and B molecules. A subvolume clearly needs both reactants present in order

for a reaction to be possible. When reaction c occurs, the product molecules are placed in the same subvolume as the reactants. If both reactants in reaction c are the same type of molecule, i.e., if $A \equiv B$, then $U_{s,B}$ in (21) is replaced with $(U_{s,A} - 1)$ because an A molecule cannot collide with itself.

The implementation of second order reactions in the microscopic regime is distinct from zeroth order and first order reactions. The stochastic behavior of second order reactions is achieved via the randomness of diffusion. Instead of defining a reaction probability, a reaction is characterized by a binding radius r_B ; see [15]. Thus, reaction c occurs if reactants A and B are separated by a distance of less than r_B after diffusing.

In general, determining the value of r_B from reaction rate k_2 is non-trivial. In [15], closed-form expressions for the binding radius are described in the limits of very small and very large time steps, i.e., when the mutual displacement of the reactants in one time step is much smaller and much larger than the binding radius, respectively. Otherwise, lookup tables are presented in the supplementary information of [15] to determine r_B for a given k_2 , time step, and *mutual* diffusion coefficient (i.e., the sum of the diffusion coefficients of the reactants).

Another complication with bimolecular reactions is when they are *reversible*. A bimolecular reaction’s reverse reaction should also define an *unbinding radius* r_U as an initial separation distance between the products. The unbinding radius is intended as a balance to ensure that the “forward” bimolecular reaction occurs at the correct frequency. Lookup tables to determine the binding radius in [15] include the unbinding radius for reversible bimolecular reactions.

In AcCoRD, the user must define the binding and unbinding radii explicitly. For a given candidate reactant molecule, we check the coordinates of every corresponding candidate reactant that is in the same region or in a neighboring region. Currently, we only apply the unbinding radius for the products of *second order* reactions so that we can model collisions between spherically-shaped molecules with non-negligible volumes. Also, since second order reactions are determined after diffusion for the current time step, any given molecule can only participate in up to *one* second order reaction in a single time step, either as a reactant or a product, so the accuracy of microscopic second order reactions is sensitive to the time step.

5. AcCoRD Algorithms and Complexity

In the previous section, we focused on modeling the individual reaction and diffusion phenomena. In this section, we describe the structure of AcCoRD’s simulation algorithms. First, we discuss the overall algorithm. Then, we comment on the computational complexity of the microscopic and mesoscopic models, including how to select appropriate values for the microscopic time step and the

mesoscopic subvolume size. Specific details of the primary simulation steps are described in detail in Appendix C.

The most significant demands on memory usage are for tracking molecules in the microscopic regime and subvolumes in the mesoscopic regime. We will discuss these tasks in greater detail in Appendix C.3 and Appendix C.4, respectively, but we introduce them here so that we can discuss their use in other stages of the simulation. Microscopic molecules are stored in linked lists, which can be easily modified as needed when molecules are created or removed. We maintain *two* lists for every type of molecule in every region. One list is for “recent” molecules that were created *within* the current microscopic time step. Each recent molecule was created at some time t_i , where $0 \leq t_i < \Delta t$, and has its own *individual* time step, i.e., $\Delta t - t_i$. The other list is for all “normal” molecules that were created before the current microscopic time step, and which do not need the additional memory for an individual time step. Generally, we add molecules to the corresponding “recent” list when they are first created and then transfer them to the “normal” list after their first diffusion step.

In the mesoscopic regime, there is a constant number of subvolumes. Thus, we maintain arrays of subvolume structures where each structure describes a single subvolume. For each subvolume, we store the identities of its neighboring subvolumes, the number of each type of molecule present, and the propensities of all potential reactions as calculated in Section 4.

5.1. Overall Algorithm

The overall AcCoRD algorithm is presented in Algorithm 1. The two input arguments are the configuration filename CONFIG and the seed number ψ for the random number generator (RNG). In line 2, the configuration text file is parsed and read to determine the simulation parameters. The initialization step in line 3 performs routine tasks such as memory allocation and initializing the RNG, but also does the following:

- Determine which regions are neighbors and which region subvolumes are neighbors.
- Determine where actors are placed relative to the regions.
- Define the network of chemical reactions.
- Initialize reaction and diffusion propensity calculations for mesoscopic subvolumes.
- Create the simulation output files.

The outer simulation loop in the AcCoRD algorithm starts at line 4 and executes the number of independent realizations specified by the configuration. At the start of each realization, the physical environment is reset to its “initial” state, i.e., with no molecules present in any region. At the end of each realization, at line 20, we write the

Algorithm 1 Overall AcCoRD Algorithm

```
1: procedure AcCoRD(CONFIG,  $\psi$ )
2:   Load configuration specified by CONFIG
3:   Initialize system
4:   for Each realization do
5:     Reset the environment
6:      $t_{\text{cur}} \leftarrow 0$ 
7:     while  $t_{\text{cur}} \leq t_{\text{end}}$  do
8:       Find next simulation step
9:       if Next step is active actor  $a$  then
10:        ACTIVE_ACTION( $a$ )
11:       else if Next step is passive actor  $a$  then
12:        PASSIVE_ACTION( $a$ )
13:       else if Next step is microscopic then
14:        MICROSCOPIC_STEP
15:       else  $\triangleright$  Next step is mesoscopic
16:        MESOSCOPIC_STEP
17:       end if
18:       Update  $t_{\text{cur}}$ 
19:     end while
20:     Write observations to file
21:     Display progress
22:   end for
23:   Cleanup
24:   return
25: end procedure
```

observations of the specified actors to the *primary simulation output file*. We then update the simulation progress to the console (as specified; the configuration file defines the frequency of progress updates). After the simulation loop, the cleanup step in line 23 writes a summary of the simulation execution to a separate *summary file* and releases all allocated memory.

The core of the overall algorithm is the evolution of the system in a single realization from time $t_{\text{cur}} = 0$ until time $t_{\text{cur}} > t_{\text{end}}$. The system evolves as a series of steps. The four possible steps are: 1) Action by an active actor; 2) Action by a passive actor; 3) Evolution of the microscopic regime; and 4) Evolution of the mesoscopic regime. These four steps are discussed in detail in Appendix C. AcCoRD has an actor-based design with a hybrid of simulation regimes, so steps do *not* occur in a predetermined order. Instead, every actor and each simulation regime has its own internal timer that defines when the corresponding step should be executed. All timers are sorted in a priority queue. The current step is determined in line 8 by the timer whose value equals t_{cur} . Once the corresponding behavior is executed, the timer is updated. Then, in line 18, the queue is re-sorted and the new time t_{cur} is determined from the new lowest timer value. This flexible design enables actors to behave at any frequency and accommodates the execution of both microscopic and mesoscopic simulation models.

5.2. Computational Complexity

We focus our discussion of AcCoRD’s computational complexity by discussing the factors that impact the execution speed. Specifically, we consider the running time of the microscopic and mesoscopic evolution steps. These two steps generally consume an overwhelming majority of a given simulation’s total run time, such that the time required to execute actor behavior and the overall AcCoRD algorithm is usually negligible. With the environment configurations that we have tested, execution speed has been considerably more of a bottleneck than the required computer memory. However, we also comment briefly on considerations for memory usage.

5.2.1. Mesoscopic Complexity

The speed of simulation in the mesoscopic regime depends on the number of events, i.e., chemical reactions within individual subvolumes or diffusion between adjacent subvolumes, and the time to execute each event. Every mesoscopic simulation step executes a single event. Each simulation has a specified end time t_{end} , so the number of events is inversely proportional to the time between individual events. The time until the next mesoscopic event is inversely proportional to the total propensity α_{T} , which is the sum of all possible individual event propensities (we described the calculation of propensities in Section 4, and we discuss the generation of event times from propensities in Appendix C.4). Thus, by inspection of the propensity equations, we can infer the event frequency and therefore simulation run time complexity for different events.

The time required to simulate diffusion in the mesoscopic regime between subvolumes of the same size can be inferred from (1), i.e., the number of diffusion events of molecule type m out of subvolume s grows with $\mathcal{O}(D_m U_{s,m}/h_s^2)$, where $\mathcal{O}(\cdot)$ refers to big O notation. Clearly, computation time will increase by increasing the diffusion rate, increasing the number of molecules, or decreasing the subvolume size. The same observation can be made by considering the diffusion between subvolumes of different sizes in (2), but from (5) the number of diffusion events from the subvolume to an adjacent microscopic region grows with $\mathcal{O}(\sqrt{D_m} U_{s,m}/h_s)$. We emphasize that the number of molecules can be time-varying, so the overall diffusion simulation speed is also time-varying.

Unlike diffusion, which always adds a molecule in one subvolume after removing it from another, the number of chemical reaction events in the mesoscopic regime depends on the net change in the total number of molecules due to each reaction. The number of *zeroth* order reactions is independent of the number of molecules, since they have no reactant. From (13), we can deduce that the number of instances of zeroth order chemical reaction c in subvolume s grows with $\mathcal{O}(k_c V_s)$.

First and second order reactions do depend on the corresponding number of molecules. From (15), we conclude

that the number of instances of first order reaction c in subvolume s with reactant m grows with $\mathcal{O}(k_c U_{s,m})$. From (21), we find that the number of instances of second order reaction c in subvolume s with reactants A and B grows with $\mathcal{O}(k_c U_{s,A} U_{s,B} / V_s)$.

Now that we have established the growth in the *number* of mesoscopic simulation events, we comment on the speed of executing a *single* event. This is described in further detail in Appendix C.4, where it is shown that a mesoscopic step takes $\mathcal{O}(\log N_{\text{Sub}} + N_{\text{Event}})$ time, where N_{Sub} is the total number of subvolumes and N_{Event} is the total number of possible events in the current subvolume (i.e., the sum of all reaction and diffusion events).

Given the above discussion, and assuming that there are no hybrid interfaces, we can write the overall complexity of the mesoscopic model run time as

$$C_{\text{meso}} = \mathcal{O} \left(t_{\text{end}} \left[\sum_{\Omega_{\text{Sub}}} \left(\sum_m \frac{D_m U_{s,m}}{h_s^2} + \sum_{k_0} k_0 V_s + \sum_{k_1} k_1 U_{s,k_1} + \sum_{k_2} \frac{k_2 U_{s,k_2,1} U_{s,k_2,2}}{V_s} \right) \right] \times (\log N_{\text{Sub}} + N_{\text{Event}}) \right), \quad (22)$$

where the summations are over all mesoscopic subvolumes, all types of molecules in subvolume s , and all zeroth, first, and second order reactions in subvolume s , respectively. U_{s,k_1} is the number of molecules in subvolume s that are the reactant associated with reaction rate k_1 , and $U_{s,k_2,1} U_{s,k_2,2}$ is the product of the corresponding numbers of reactants associated with reaction rate k_2 . t_{end} is the simulation end time, assuming that it starts at $t = 0$. If hybrid interfaces are present, then we must include in (22) a summation of $\sqrt{D_m} U_{s,m} / h_s$ over molecule types for the subvolumes that are adjacent to a microscopic region.

From (22), using more subvolumes will increase both the number of simulation events and the time to execute each event. Thus, one factor in the selection of subvolume size is the simulation run time. However, it is not the only factor, and in fact there are practical bounds that constrain subvolume size for an accurate simulation, particularly if bimolecular reactions can occur. For example, as defined in [31, Eq. (7)], a 3D subvolume will only be “well-mixed” (such that the second order reaction propensity in (21) is accurate) if the subvolume length h_s is much less than $\sqrt{6Dt_c}$, where t_c is the characteristic time of the fastest bimolecular reaction. Alternatively, if subvolumes are too small, then bimolecular reaction propensities will depend on reactant molecules in neighboring subvolumes. A lower limit on 3D subvolume size for a given bimolecular reaction was calculated in [59] to be about $3.2r_B$, where r_B is the binding radius of the two reactants. For subvolumes smaller than this limit, corrections are needed to modify the bimolecular reaction propensity, but such changes are outside the scope of this work.

5.2.2. Microscopic Complexity

Unlike the mesoscopic regime, the frequency of microscopic steps in simulation time does not vary over the course of the simulation. Instead, the number of microscopic simulation steps is explicitly defined by the constant simulation time step Δt as $\lceil t_{\text{end}} / \Delta t \rceil + 1$, where $\lceil \cdot \rceil$ is the ceiling function. Generally, a smaller time step increases the simulation accuracy, but as with subvolume size there are practical limits to consider. In order to assume Brownian motion, the time step must be large enough to ignore hydrodynamic memory effects, which occur over time scales less than $t = \gamma^2 \rho / \eta$, where γ is the diffusing molecule radius, ρ is the fluid density, and η is the fluid viscosity⁴; see [60]. For time steps that are too large, interactions between a molecule and environmental features (especially surfaces and other boundaries) will be modeled inaccurately, since we generally consider straight line diffusion trajectories. A simple bound is to consider a molecule’s root mean square distance along each dimension in one time step, i.e., $\sqrt{2D\Delta t}$. This distance should be much smaller than the resolution of the smallest environmental features.

As we discuss in Appendix C.3, a microscopic simulation step has separate stages for zeroth order reactions, non-surface first order reactions, diffusion (with surface reactions), and second order reactions. From the run time complexity results in Appendix C.3, we can write the overall complexity of the microscopic model run time as

$$C_{\text{micro}} = \mathcal{O} \left(\frac{t_{\text{end}}}{\Delta t} \sum_{\Omega_{\text{micro}}} \left[\sum_{k_0} k_0 V_r + \sum_{k_1} k_1 N_{k_1,r} + N_{\text{micro},r} |\Omega_{\text{neigh},r}| + \sum_{k_2} N_{k_2,1,r} \left(N_{k_2,2,r} + \sum_{w \in \Omega_{\text{neigh},r}} N_{k_2,2,w} \right) \right] \right), \quad (23)$$

where each summation is over the set of microscopic regions, the set of corresponding chemical reactions (i.e., zeroth, first, and second order with rates k_0 , k_1 , and k_2 , respectively), and the set of regions $\Omega_{\text{neigh},r}$ that neighbor region r . V_r is the region’s volume, $N_{k_1,r}$ is the number of reactant molecules for the first order reaction associated with rate k_1 , $N_{\text{micro},r}$ is the number of molecules in region r , and $|\Omega_{\text{neigh},r}|$ is the total number of regions that are neighbors to region r . $N_{k_2,1,r}$ and $N_{k_2,2,r}$ are the numbers of first and second reactant molecules in region r for the second order reaction associated with rate k_2 , and $N_{k_2,2,w}$ is the corresponding number of second reactant molecules in neighboring microscopic region w . The term $N_{\text{micro},r} |\Omega_{\text{neigh},r}|$ represents the cost for diffusing each

⁴Using typical values for water density (10^3 kg/m^3) and viscosity ($10^{-3} \text{ kg/(m}\cdot\text{s)}$), and molecule radii on the order of less than 10 nm, hydrodynamic memory effects would occur on a time scale of less than 1 ns, which is much less than the smallest microscopic time steps considered in this paper.

molecule and verifying its trajectory, including testing for entering the mesoscopic regime.

We can compare the complexity in computational run time of the mesoscopic and microscopic models by comparing (22) and (23). Both rely on the characteristic parameters that define each regime, i.e., time step in the microscopic model and subvolumes in the mesoscopic model. However, they show similar sensitivity to all three orders of chemical reactions. We will gain more specific insights into the time for simulating diffusion by measuring simulation run times in Section 7.

5.2.3. Memory Usage

Finally, we comment on memory usage. AcCoRD’s greatest memory demands are to store information about the individual subvolumes and the individual molecules. Subvolumes are used in both the mesoscopic and microscopic regimes, although more memory is used for each subvolume in the mesoscopic regime in order to: 1) count the number of each type of molecule in the subvolume, and 2) track the subvolume’s place in the indexed priority queue to determine when its next event will occur. Memory for individual molecules is needed in microscopic regions, and also to store molecule locations (from both regimes) before a realization’s observations are written to the output file. Generally, if too much memory is required, then we can either represent mesoscopic regions with fewer subvolumes or have fewer molecules in the microscopic regime.

6. Using AcCoRD

In this section, we describe the general work flow for an AcCoRD user. This section does not provide step-by-step instructions; a user can refer to the online code documentation for specific details on the latest version; see [21]. The online documentation includes installation and usage instructions, descriptions of all configuration options, and many sample configuration files. We described the format of a configuration file in Section 3. Here, we focus on AcCoRD’s usability, and we intend for this section to accommodate future updates to the code.

6.1. Preparing a Simulation

Every simulation is based on the environment and behavior specified in a configuration file. The sample configuration files are intended to be copied and renamed for modification. To confirm that the environment is designed as intended (i.e., with the correct size and relative placement of objects), the user can draw the regions and actors via a configuration plotting function in MATLAB. For example, *every* figure with a sketch in this paper (with the exception of Fig. 6) was prepared using an AcCoRD configuration file and the plotting function.

The AcCoRD simulator is compiled as a single executable file. Executables and build scripts are currently

provided for Windows and Linux operating systems. The executable can be called directly from a command line interface, and it is also possible to run the simulator remotely on a computing cluster (such as those operated by Compute/Calcul Canada, which were used for many of the simulations in Section 7). Providing a seed for the random number generator at run time will over-ride the seed defined in the configuration file. The intent is that a user can distribute the computing load of a simulation by running different realizations on different processors; providing a unique seed in each call will avoid repeated realizations.

6.2. Running a Simulation

To facilitate automated or remote execution, a simulation normally runs without any input from the user. The command line interface displays a brief summary of the configuration, including the number of regions, actors, and subvolumes. It also shows the relative directory where the simulation output is being written. We use a timer to measure the duration of every realization and provide updates on the estimated time remaining under the assumption that each realization will execute in comparable time. The total simulation time, i.e., the time required to execute the for-loop in Algorithm 1 on page 17, is also displayed at the end.

AcCoRD can return warnings or errors at run time. In general, we use warnings to indicate invalid or missing configuration information that can be replaced by hard-coded default values. We use errors to deal with memory allocation problems or identify invalid configuration information that is too late to automatically correct (for example, regions that are nested improperly). Warnings and errors are described in the command line with corresponding details, such as the name of the region, actor, or reaction that led to the warning or error. By default, the creation of any warnings during the loading of the configuration will pause the simulation and prompt the user to continue (especially since the “corrected” configuration may not behave as intended or may lead to other errors). Any errors will terminate the simulation.

6.3. Post-Processing

Running a simulation once, i.e., with one seed value, generates *two* output files that are both labeled with the corresponding seed number. A *summary* file is in JSON-format and it describes the information in the custom-formatted *primary* output file. This information includes how many actors were being recorded and how much data is associated with each actor. All post-processing utilities were developed in MATLAB and the user can load simulation output with an import function. By specifying multiple seed values, multiple pairs of output files can be imported and aggregated simultaneously. For example, a simulation that was repeated 10 times (with 10 different seeds) and simulated 1000 realizations with each seed can be imported as a single simulation with 10^4 realizations and saved to a MATLAB mat-file.

We have a number of visualization tools for AcCoRD’s simulation output. Current options include the following:

1. Visualizing individual molecules in an animation or video. We combine the configuration plotting tool, which displays the specified regions and actors, with plots of individual molecules at locations observed by passive actors within a *single* realization. A user generally has full control over what simulation components are displayed, what time interval is viewed, how the camera is controlled (can be static or dynamic), and whether to create a series of figures or an exported video file. Custom overlay information can include a simulation progress timer or a display of the number of molecules of some type observed by one of the passive actors. A total of eight videos are included in [23].
2. Plotting the time-varying signal as a curve. Curves can correspond to individual realizations or be averaged over any subset of realizations in the imported simulation file. It is also possible with the same syntax to add curves of user-defined data, so a user can add analytical results or other information.
3. Plotting the signal’s statistical distribution as a probability mass function (PMF) or cumulative distribution function (CDF). These empirical distributions are determined from a specified subset of simulation realizations, and can be time-varying (i.e., drawn as a 3D surface) or averaged over multiple observation times (i.e., drawn as a 2D curve). Analogous to plotting the time-varying signal, we can also plot analytical distributions based on a user-defined trial probability (i.e., probability of observing a given molecule) and number of trials (i.e., number of molecules). A distribution can be Binomial or either the Poisson or Gaussian approximation of the Binomial distribution. We review these distributions in Appendix D.
4. Plotting the signal’s time-varying mutual information relative to one or more reference observations in 2D and 3D, respectively. Mutual information can be used as an indicator of observation independence, as we considered in [61]. Consecutive observations that are not sufficiently separated in time will be identical and have mutual information equal to the entropy of the signal. Observations that are sufficiently separated in time should be independent and have zero mutual information. Since “zero” mutual information cannot be measured directly from a finite number of samples (see [62]), a user can also plot *Monte Carlo mutual information*, which measures the mutual information between finite samples of *independent* Binomial random variables. We review the calculation of mutual information in Appendix D.

Communications performance metrics such as the bit error rate (BER) depend on the specific design of the re-

ceiver’s detector. AcCoRD does not impose a specific detector design. By making the receiver observations available for post-processing in MATLAB, a user can readily implement their own detector off-line and measure its performance. Utilities that implement common detector designs will be added in a future update. Also, the future implementation of dependent actors will include detectors as part of the online simulation.

7. AcCoRD Simulation Results

In this section, we present a series of simulation results generated by AcCoRD. Our overall aim is to demonstrate the functionality and the accuracy of the software, including its design as a “sandbox” reaction-diffusion solver and its generation of output that is suitable for signal processing (including communications analysis). All results were generated using version 0.7 or 0.7.0.1 and can be equivalently generated by version 1.0. Some results have accompanying videos in [23] to show a sample of the evolution of the environment with a reduced number of molecules. These videos are also summarized in Appendix E. Generally, we have avoided focusing on results that we have already presented in preliminary work with earlier versions of AcCoRD. These earlier results include a study on the choice of statistical distribution to describe the number of observed molecules in [26], a demonstration of a system with a large number of molecule sources in [63], a study on the accuracy of the assumption that a transmitter is a point source in [25], and a direct comparison between passive and absorbing receivers in [64]. The reader may refer to these earlier works for results from those specific scenarios.

We consider four system environments in this section, each with a series of variations. These systems are representative of AcCoRD’s features but in consideration of space they are not comprehensive⁵. We now summarize each system and its contribution to our study. There is at least one video in [23] for each system.

System 1 is a bounded rectangular environment, as shown in Fig. 8 and also in Fig. 1 on page 3. In Section 7.1, we use System 1 to study the accuracy of using a hybrid of microscopic and mesoscopic simulation models for a diffusion-only environment. We also verify the accuracy of (3) to describe the transition rate between mesoscopic subvolumes of different sizes. We measure the distribution of molecules observed in different parts of the system, the mutual information between consecutive observations at a given location, and the simulation run time.

System 2 is a long rectangular rod with a perfectly-absorbing surface at one end, as shown in Fig. 2 on page 3.

⁵Specifically, we do not simulate a reversible first order reaction in solution or use bimolecular reactions to model molecule crowding. However, sample configurations for these cases are included with the source code in [21]. We also only consider “steady state” reaction probabilities for surface reactions that have finite reaction rates.

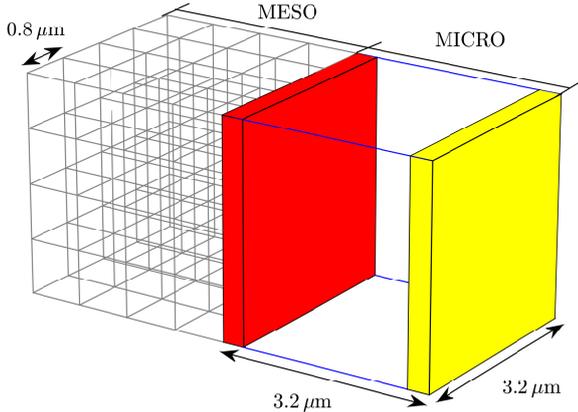


Figure 8: Layout of the hybrid environment in System 1. Both regions are cubes. The variation shown has a mesoscopic region (MESO) with subvolumes of width $h = 0.8 \mu\text{m}$. The red and yellow areas of the microscopic (MICRO) region have a depth of $0.32 \mu\text{m}$ and are observed by passive receivers. A sample video of the simulation of this environment is available as Video 2 in [23].

System 2 is used as an example of a hybrid environment with both diffusion and chemical reactions. In Section 7.2, we compare the accuracy of the fully microscopic model with that of the hybrid model at different locations along the rod. We vary the location of the hybrid interface and also measure the simulation run time.

System 3 has two large concentric spherical surfaces and places molecules between them, as shown in Fig. 3 on page 4. Every molecule can either adsorb to one of the surfaces or transition through it (i.e., the surface is a membrane). The size of this system makes it locally 1D so that it is analogous to the environment considered in [24, Fig. 6a]. Thus, this system is used in Section 7.3 to verify our implementation of the surface reaction probabilities that were derived in [24] (and summarized in Appendix B), while also verifying the accuracy of the collision detection and reflection off of spherical surfaces (which are geometric problems whose details are outside the scope of this paper).

Finally, System 4 considers a point molecule source transmitter and a spherical receiver, as shown in Fig. 4 on page 4. This system and its variations have been commonly studied in the MC literature, cf. [13, 65, 66, 67]. In Section 7.4, we consider the channel impulse response at a passive receiver when the system is unbounded versus having an absorbing or reflecting outer boundary. We also consider the impulse response with degradation while diffusing. To do so, we compare a first order degradation reaction with enzyme-mediated degradation via Michaelis-Menten kinetics (see [11, Ch. 10]). We measure the time-varying PMF at a passive receiver when the transmitter releases a series of finite pulses of molecules. We also verify the impulse response when the receiver has a fully- or partially-absorbing surface.

7.1. System 1: Hybrid and Multi-Scale

The layout of System 1 is a box as shown in Fig. 8, with a width and depth of $3.2 \mu\text{m}$ and a length of $6.4 \mu\text{m}$. We first divide the environment into two cubic halves, where one half is microscopic with a time step of $\Delta t = 0.5 \text{ ms}$ and the other half is mesoscopic with subvolumes of a uniform size that we vary from $h = 0.1 \mu\text{m}$ to $h = 3.2 \mu\text{m}$. We place two observers in the microscopic region (one at the hybrid interface and the other at the opposite end of the region; shown in red and yellow, respectively, in Fig. 8), and each is a passive actor with a depth of $0.32 \mu\text{m}$ so they observe 10% of the microscopic region, respectively. An active actor uniformly initializes 1000 molecules with a diffusion coefficient of $D = 10^{-10} \frac{\text{m}^2}{\text{s}}$ throughout the entire system at time $t = 0$, and the passive actors observe the system at intervals of 0.01 s for 1 s (i.e., 100 observations). With these parameters, the average diffusion distance of a microscopic molecule along one dimension in one time step is $\sqrt{2D\Delta t} \approx 0.32 \mu\text{m}$. Thus, the range of subvolume sizes include subvolumes that are relatively small (i.e., $h \sim \sqrt{D\Delta t}$) and relatively large (i.e., $h \gg \sqrt{D\Delta t}$), so we can compare rules (9) and (10) when determining the tangential placement of molecules that enter the microscopic region from a mesoscopic subvolume. We define $l_{\text{max}} = 10 \mu\text{m}$, so no molecules in the microscopic region are excluded from the possibility of entering and exiting the mesoscopic region during a diffusion step.

We seek to get a sense of the trade-offs of the hybrid environment, including its speed and accuracy, when simulating diffusion. In this system, the hybrid interface between the microscopic and mesoscopic regions is relatively large compared to the total size of the environment. Thus, we expect that the accuracy will be very sensitive to that at the interface, and that the simulation run time will be very sensitive to the overhead introduced by having the interface. The total simulation time of 1 s is long enough a molecule's expected displacement along each dimension to be a few times larger than the entire system, so any long term bias in the diffusion between regions should be observable.

In Fig. 9, we simulate each value of h 10^4 times and plot the probability mass function of all 100 observations at each observer (as a single PMF for each observer constructed from 10^6 observations). We separately consider the transition rules (9) or (10), i.e., assume that the mesoscopic subvolumes are small or large, respectively. The hybrid environments are compared with a variation where the entire system is microscopic (i.e., where there is no hybrid interface). Video 2 in [23] shows a sample realization when $h = 0.8 \mu\text{m}$.

Given the uniform molecule distribution, we expect that each PMF in Fig. 9 should have a peak value at about 50 molecules. This is the case for the benchmark microscopic system. We observe that all hybrid variations in Fig. 9 underestimate the actual number of molecules close to the hybrid interface (see subplots (a) and (b)). We see

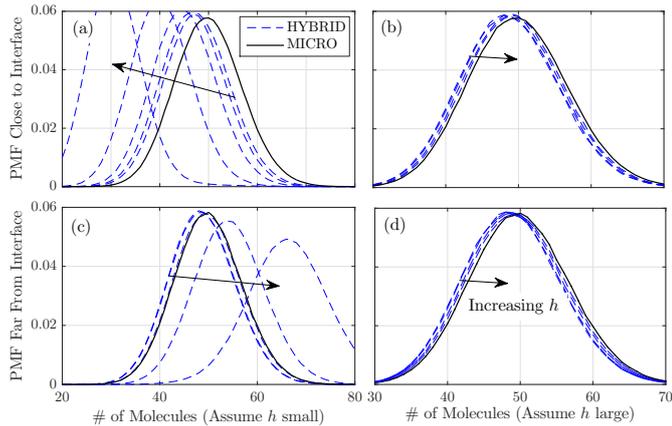


Figure 9: Probability mass functions of number of molecules observed in subregions of the hybrid environment in System 1. We observe 10% of the microscopic region closest to the hybrid interface (in (a) and (b)) and 10% of the microscopic region furthest from the hybrid interface (in (c) and (d)). The hybrid transitions are based on the assumption that the mesoscopic subvolumes are small (i.e., that $h \sim \sqrt{D\Delta t}$, in (a) and (c)) or that they are large (i.e., that $h \gg \sqrt{D\Delta t}$, in (b) and (d)) relative to the average diffusion distance ($\sqrt{2D\Delta t} \approx 0.32 \mu\text{m}$) in the microscopic region. The arrows are in the directions of the PMFs for increasing mesoscopic subvolume size h . We plot $h = \{0.2, 0.4, 0.8, 1.6, 3.2\} \mu\text{m}$, except we omit the $h = 3.2 \mu\text{m}$ case in (b) and (d) because its PMFs are indistinguishable from those when $h = 1.6 \mu\text{m}$. Simulations with $h = 0.1 \mu\text{m}$ were also performed, but the resulting PMFs are indistinguishable from those when $h = 0.2 \mu\text{m}$ so they are not shown.

that accuracy in subplot (a), i.e., when we assume that h is small, improves with decreasing h . The accuracy in subplot (b), i.e., when we assume that h is large, improves with increasing h . However, none of the PMFs of the observer close to the interface match the microscopic benchmark well when we assume that h is small. This inaccuracy is most likely due to the hybrid transition rule for small h , which does not account for the environment boundary when molecules leave the mesoscopic regime. This can be corrected in a future release of AcCoRD. The accuracy is better when we consider the observer far from the interface (subplot (c)), but is still not as accurate as the corresponding case when we assume that the subvolumes are large (subplot (d)). Again, this can be improved by accounting for molecules crossing the environment boundary when leaving the mesoscopic regime. Currently, the accuracy appears to be much less sensitive to the subvolume size when we assume that the subvolumes are large, so we will only consider this assumption and use (10) for the remainder of our results in this paper.

In Fig. 9, we collected all observations by a passive actor into a single PMF and we did not consider the impact of the observation times. It is interesting to consider

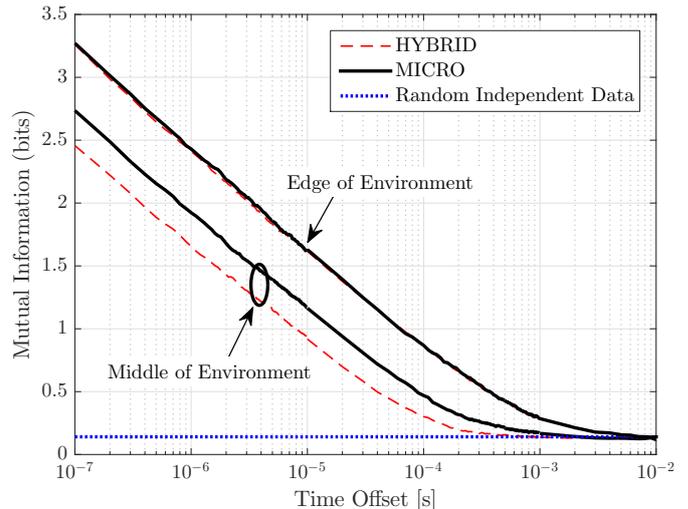


Figure 10: Mutual information between observations as a function of the time between the observations. The hybrid environment in System 1 with $h = 0.8 \mu\text{m}$ is compared with the non-hybrid (i.e., microscopic) version. Mutual information between an equivalent number of randomly-generated independent observations is shown as a lower bound.

the impact of the hybrid interface on the joint observation statistics, since communications analysis and related signal processing often relies on assuming independent observations; see [61, 68]. We address this idea in Fig. 10, where we measure the mutual information using (D.14) between consecutive observations as a function of time between when the observations were taken. We consider the first observation as the reference time and vary the offset from 10^{-7} s to 10^{-2} s , i.e., most of the offsets shown are smaller than the microscopic time step $\Delta t = 0.5 \text{ ms}$ used in Fig. 9. The mutual information for each offset is calculated from 10^4 simulated observations, and we also compare with the mutual information measured from 10^4 pairs of independently-generated random variables with the same mean (i.e., 50) as a bound of “true” independence⁶. We use base 2 for the logarithm in (D.14) so that mutual information is measured in bits.

In Fig. 10, the only hybrid variation that we consider is when $h = 0.8 \mu\text{m}$, but similar results can be observed with other subvolume sizes. When the observer is *far* from the interface, i.e., at the edge of the environment, the mutual information between observations is identical to that when the entire system is microscopic. However, these observations are not truly independent until there is about 10 ms between them, i.e., more than an order of magnitude longer than the microscopic time step in Fig. 9. In this case, one cannot automatically assume that observations

⁶As we discuss in Appendix D, independent variables actually have zero mutual information, but calculating mutual information from a finite number of realizations generally results in a non-zero value; see [62]. We use such a value here as an independence bound.

taken more frequently than every 10 ms will be independent.

There are two interesting observations about the mutual information at the observer in the *middle* of the environment, i.e., at the interface in the hybrid case. First, the mutual information in both system variations is *less* than it is at the edge of the environment for the same offset. This is because more molecules at the edge reflect off of the system boundary and remain within the observer, so there are fewer molecule transitions into and out of the observer and hence less uncertainty about the values of consecutive observations. Second, the mutual information in the hybrid case is less than in the microscopic system; the region transitions at the hybrid interface introduce *additional uncertainty* in the molecule locations. Determining the precise cause of this uncertainty at the interface is an interesting problem for future work. Nevertheless, we observe that the mutual information between observations in the middle of the environment tends towards that measured for randomly-generated independent observations for both simulation models.

It is also of interest to consider variations of System 1 that are entirely mesoscopic, and also to use subvolumes of different sizes. We consider using mesoscopic subvolumes of size $h = 0.4 \mu\text{m}$ throughout the entire system, as well as a variation where half of the environment is replaced with subvolumes of size $h = 1.6 \mu\text{m}$, and compare the corresponding observation PMFs in the middle of the system with that of the microscopic benchmark in Fig. 11. All three of these PMFs are practically indistinguishable, which also verifies the transition rule that we derived for mesoscopic subvolumes of different sizes in (3). Comparable results can be observed using mesoscopic subvolumes of other sizes.

Now that we have considered the accuracy of microscopic, mesoscopic, and hybrid diffusion, we briefly consider the simulation run times in Fig. 12. We measure the average run time per realization as determined by running at least 100 realizations on an Intel i7 desktop PC. The run time of the fully mesoscopic system is clearly proportional to the number of subvolumes, as expected from the run time complexity expression in (22), and is many orders of magnitude faster than the microscopic system when the subvolumes are very large (e.g., in the limiting case of $h = 3.2 \mu\text{m}$, there are only two subvolumes). However, the microscopic system is faster than the mesoscopic system when the subvolumes are $0.4 \mu\text{m}$ in size or smaller. As expected, the run time of the multi-scale mesoscopic case, which divides each half of the environment into subvolumes of size $h = \{0.4, 1.6\} \mu\text{m}$, is between those when the environment has subvolumes of only one of those sizes. The run times in the hybrid variations are generally no faster than when the entire system is microscopic. They become faster than the mesoscopic system when the subvolumes are smaller than $0.4 \mu\text{m}$. The run times are not an average of the underlying microscopic and mesoscopic run times, since the hybrid system also introduces a computational

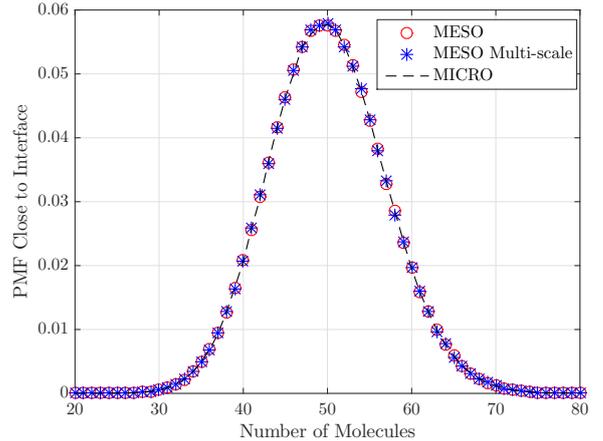


Figure 11: Probability mass functions of number of molecules observed in a subregion in the middle of System 1 (5% of the total size). We model the environment as fully microscopic, fully mesoscopic with subvolumes of size $h = 0.4 \mu\text{m}$, or as fully mesoscopic where half of the subvolumes have size $h = 0.4 \mu\text{m}$ and the other half have size $h = 1.6 \mu\text{m}$ (“MESO Multi-scale”). The corresponding probability mass functions are practically indistinguishable.

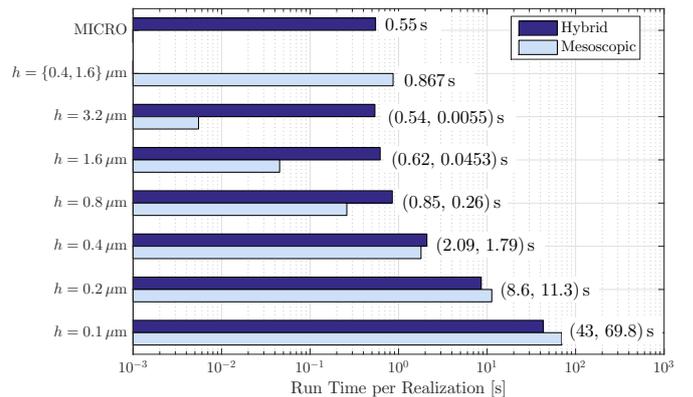


Figure 12: Simulation run times for the variations of System 1. The hybrid variations measured were those assuming that the subvolume size h was large. Run time was averaged over at least 100 realizations of each variation on an Intel i7 desktop PC. The run time of the multi-scale mesoscopic variation, with subvolumes of size $h = \{0.4, 1.6\} \mu\text{m}$, is between that of the corresponding mesoscopic variations with a uniform subvolume size.

overhead to manage the transitions between the two simulation models, as discussed in Appendix C.3 (especially with such a large value of $l_{\text{max}} = 10 \mu\text{m}$). We will consider the potential benefits of the hybrid model in further detail when we study System 2.

Finally, for completeness, we consider the PMFs of the common statistical distributions using the parameters of System 1. Specifically, in Fig. 13 we plot the binomial PMF for 1000 trials that each have a success probability

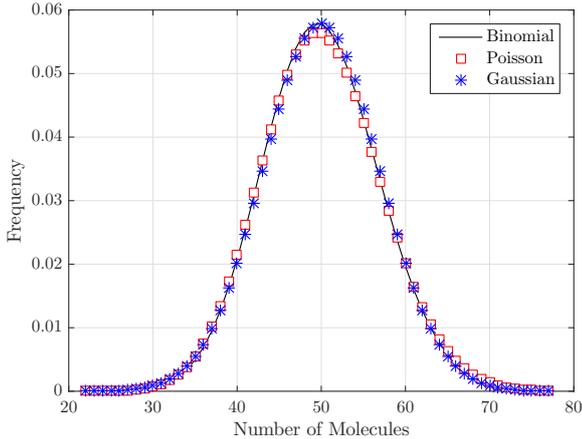


Figure 13: Probability mass function of a Binomial random variable with 1000 trials and a trial success probability of 5%. This distribution, which matches the expected behavior of the observations in System 1, is also compared with the Poisson and Gaussian approximations of the Binomial. Both approximations show slight deviations from the Binomial, particularly near the peak.

of 5% (since there are 1000 molecules that have an unconditional probability of 5% of being observed by a given passive actor), and the Poisson and Gaussian approximations of that PMF. The Binomial PMF is identical to that of the microscopic system (which we do not plot again in Fig. 13; refer to Figs. 9 and 11). Both approximations are close to the Binomial PMF in this case, but neither is indistinguishable; the Poisson approximation underestimates the likelihood of the peak observation, whereas the Gaussian approximation underestimates and then overestimates the likelihood of the observations with values less than and greater than 50, respectively. We discussed the general accuracy of these approximations further in [26].

7.2. System 2: Hybrid Reaction-Diffusion

In System 2, a portion of which is shown in Fig. 2 on page 3, we demonstrate the potential gain in computation speed by defining an environment with both microscopic and mesoscopic regions. As with System 1, we consider a simple hybrid environment, but this one has a surface reaction in the microscopic regime and (when applicable) a much larger mesoscopic regime. System 2 is a 0.5 mm long and $5 \mu\text{m}$ wide rectangular rod with one *perfectly*-absorbing end; i.e., every molecule that crosses that end is absorbed. 1000 molecules with diffusion coefficient $D = 10^{-10} \frac{\text{m}^2}{\text{s}}$ are initially placed uniformly over the entire rod, and we observe the number of molecules over three $5 \mu\text{m}$ sections of the rod. The sections are at distances of 0, 5, and $20 \mu\text{m}$ from the absorbing end.

The area closest to the absorbing surface is modeled as a microscopic region with a time step of $\Delta t = 1 \mu\text{s}$, which we found to be small enough to accurately model the perfect absorption reaction at the surface. We first

consider the rod modeled as a single microscopic region, and then variations where we model most of the rod as a mesoscopic region with subvolumes of width $h = 5 \mu\text{m}$ (such that the condition $h \gg \sqrt{D\Delta t}$ is easily satisfied). In the hybrid variations, the hybrid interface is placed at $d_i = \{5, 10, 25, 50\} \mu\text{m}$ from the absorbing surface, such that the environment is either 1%, 2%, 5%, or 10% microscopic, respectively. A fully mesoscopic variation was not possible, since AcCoRD cannot yet simulate surface reactions in the mesoscopic regime. Due to the very small microscopic time step, we set $l_{\text{max}} = 1 \mu\text{m}$ to limit the proximity of microscopic molecules to the hybrid interface that are tested for entering and exiting the mesoscopic region within a diffusion step.

We simulated every variation 1000 times and plot the average time-varying signal at each observer in Figs. 14 and 15, where we focus on the hybrid interface being close to and far from the absorbing end, respectively. We measure the average realization run time for all variations in Fig. 16. Video 3 in [23] shows a sample realization when $d_i = 25 \mu\text{m}$.

The expected time-varying signal in this environment can be derived in closed form, *if* we assume that the rod has infinite length. Using the point concentration defined in [7, Eq. (3.13)], and the integral of the error function in [57, Eq. (4.1.1)]

$$\int \text{erf}(y) dy = y \text{erf}(y) + \frac{1}{\sqrt{\pi}} \exp(-y^2), \quad (24)$$

then it can be shown that the expected time-varying *concentration* is

$$C(t) = C_0 \left[r_2 \text{erf} \left(\frac{r_2}{2\sqrt{Dt}} \right) - r_1 \text{erf} \left(\frac{r_1}{2\sqrt{Dt}} \right) + 2\sqrt{\frac{Dt}{\pi}} \left(\exp \left(-\frac{r_2^2}{4Dt} \right) - \exp \left(-\frac{r_1^2}{4Dt} \right) \right) \right], \quad (25)$$

where C_0 is the initial concentration and r_1 and r_2 are the distances from the absorbing surface to the start and end of the observer, respectively. In System 2, the initial linear distribution of molecules is 1000 molecules per $500 \mu\text{m}$, so we multiply the time-varying concentration in (25) by the observer length of $5 \mu\text{m}$ to get the time-varying number of molecules at the observer. We apply this scaling of (25) for the “Analytical” curves in Figs. 14 and 15.

Both Figs. 14 and 15 demonstrate that the observations in the fully microscopic system are consistent with the analytical curves. Thus, System 2 is sufficiently long to assume that it has infinite length for the time scale considered. In Fig. 14, we see that the hybrid model with the interface $5 \mu\text{m}$ from the surface (i.e., $d_i = 5 \mu\text{m}$) overestimates the number of molecules counted by the observer adjacent to the absorbing surface, and significantly underestimates the number of molecules counted by the observer $5 \mu\text{m}$ from the surface. In this environment, since

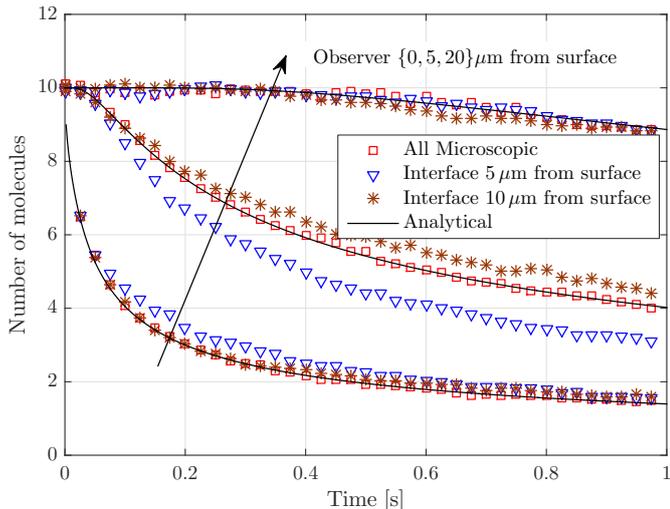


Figure 14: Average time-varying number of molecules observed by each observer in System 2. The hybrid variations place the interface either $d_i = \{5, 10\} \mu\text{m}$ from the absorbing surface.

$d_i = 5 \mu\text{m}$, these two observers are adjacent to the hybrid interface. The interface’s proximity to the absorbing surface is most likely the cause of these deviations, since the hybrid interface transition rules were originally derived in [39] under the assumption that there were no molecule “sinks” in the vicinity. There is much less deviation in this variation at the observer $20 \mu\text{m}$ from the surface, which is sufficiently far from the interface to not be affected by the inaccuracies that it introduces over the time scale that we simulated.

When the hybrid interface is $d_i = 10 \mu\text{m}$ from the surface, then the accuracy of the molecules observed at $0 \mu\text{m}$ and $5 \mu\text{m}$ from the interface are more accurate than when $d_i = 5 \mu\text{m}$. However, this variation is actually less accurate for the observer at $20 \mu\text{m}$ from the surface, particularly after time $t = 0.5\text{s}$. This time is how long it takes for a molecule to travel an average diffusion distance of $10 \mu\text{m}$ along one dimension, which is the distance from the hybrid interface to this observer, so the accuracy at the interface is a factor. Overall, placing the interface at $d_i = 10 \mu\text{m}$ gives us better (albeit still imperfect) accuracy near the absorbing surface, at the cost of worse accuracy further from the surface over the time scale shown (i.e., where the average diffusion distance for one molecule is about $14 \mu\text{m}$).

In Fig. 15, we confirm whether the trade-offs in local accuracy are dominated by proximity to the hybrid interface. For the time scale simulated, this is apparently not the case. When the hybrid interface is either $d_i = \{25, 50\} \mu\text{m}$, the accuracy is comparable to that of the fully microscopic variation. This is even the case when $d_i = 25 \mu\text{m}$ and the observer is $20 \mu\text{m}$ from the surface, i.e., when the observer is adjacent to the interface. So, we maintain our claim that the dominant factor affecting accuracy in the hybrid sys-

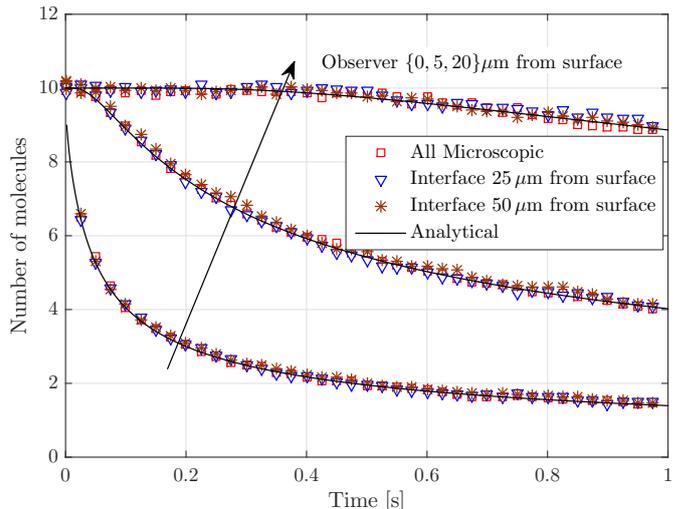


Figure 15: Average time-varying number of molecules observed by each observer in System 2. The hybrid variations place the interface either $d_i = \{25, 50\} \mu\text{m}$ from the absorbing surface.

tem is the interface’s proximity to the absorbing surface, although further analysis would be needed to confirm this analytically.

We complement our study of the accuracy of the System 2 variations by considering the simulation run times in Fig. 16. We observe that placing the hybrid interface $d_i = 5 \mu\text{m}$ from the surface, which made the system only 1% microscopic, improved the simulation run time by over 2 orders of magnitude. Even making the system 5% microscopic, which corresponded to a negligible loss in accuracy, still improved the run time by over an order of magnitude. From these results, and those observed for System 1 in Fig. 12, we gain some insight into when it is appropriate to use a hybrid model. We make the following general observations:

- The hybrid interface introduces a computational overhead, as discussed in Appendix C.3, since we need to check microscopic molecules for entering and leaving the microscopic regime within a single simulation time step. Thus, it is beneficial if the size of the interface is small relative to the larger of the adjacent regions. This was true for System 2 but not for System 1.
- The hybrid interface is beneficial if the simulation can take advantage of the benefits of both simulation models. In System 2, the microscopic model needed a very small simulation step to accurately simulate the surface reaction (which AcCoRD cannot currently simulate in the mesoscopic regime). However, the mesoscopic model simulates events at the frequency in which they locally occur. Diffusion “far” from a reactive surface does not need to be modeled with a small time step, and in System 2 the

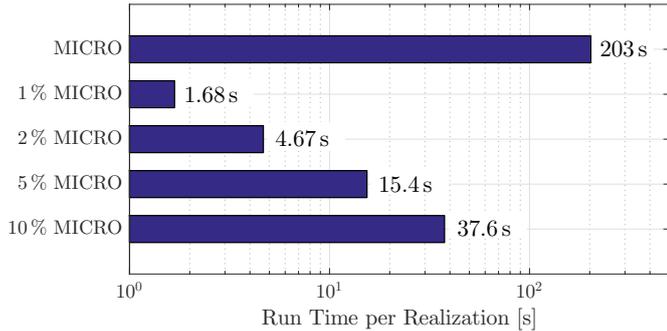


Figure 16: Simulation run times for the variations of System 2. The distances to the hybrid interface are represented via the relative size of the microscopic region, such that the rod is $\{1, 2, 5, 10\}$ % microscopic when the interface is $d_i = \{5, 10, 25, 50\}$ μm from the absorbing surface, respectively. Run time was averaged over at least 100 realizations of each variation on an Intel i7 desktop PC.

hybrid model took advantage of this. We also note that, in future work, a microscopic model that uses different time steps in different regions may provide a similar speed benefit, although the hybrid model could still have better memory usage by not needing to model each molecule in the mesoscopic regime.

- A user placing a hybrid interface should consider its proximity to reactive surfaces or other environmental features (e.g., unusual boundary shapes) that are not accounted for in the hybrid interface transition rules.

7.3. System 3: Reactive Spherical Surfaces

In System 3, we consider an approximation of the environment considered in [24, Fig. 6a] to verify the surface reaction probabilities derived in [24] (and which we summarize in Appendix B). However, instead of the 1D environment considered in [24], System 3 has two concentric spherical surfaces with radii $120 \mu\text{m}$ and $122 \mu\text{m}$, as shown in Fig. 3 on page 3. This system is locally 1D, and we can also use it to verify our implementation of reflections off of spherical surfaces. This environment is fully microscopic and we use a time step of $\Delta t = 1 \text{ ms}$.

For consistency with [24], we consider 3 types of molecule: 1) Type I can be irreversibly absorbed by the outer surface and reflects off of the inner surface; 2) Type II can reversibly adsorb to the outer surface and reflects off of the inner surface; and 3) Type III can reversibly transition across the inner surface and reflects off of the outer surface. We separately consider both “slow” and “fast” reaction kinetics for each type of molecule, using the reaction coefficients listed in Table 2, and we apply the “steady state” surface reaction probabilities defined in Appendix B. All molecules have a diffusion coefficient of $5 \times 10^{-12} \frac{\text{m}^2}{\text{s}}$, and each simulation begins by uniformly placing 20000 molecules of the given type between the

Table 2: Surface reaction coefficients used by System 3. “IRR” and “REV” stand for irreversible and reversible, respectively.

Reaction	Slow Reaction	Fast Reaction
IRR Absorption	$4.23 \frac{\mu\text{m}}{\text{s}}$	$85.9 \frac{\mu\text{m}}{\text{s}}$
REV Adsorption	$4.23 \frac{\mu\text{m}}{\text{s}}$	$85.9 \frac{\mu\text{m}}{\text{s}}$
Desorption	28 s^{-1}	276 s^{-1}
Membrane	$4.36 \frac{\mu\text{m}}{\text{s}}$	$5045 \frac{\mu\text{m}}{\text{s}}$

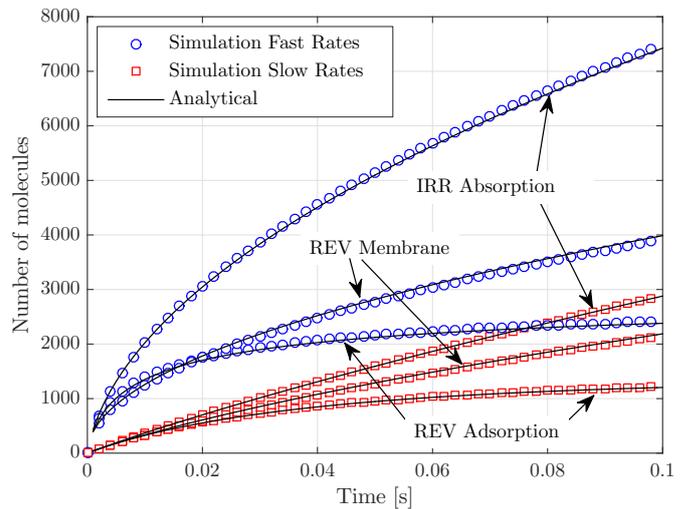


Figure 17: Average time-varying number of molecules absorbed/adsorbed to the outer surface (“Absorption/Adsorption”) or within the inner surface (“Membrane”) in System 3. This environment is a spherical analog to the 1D system studied in [24, Fig. 6a]. “IRR” and “REV” stand for irreversible and reversible, respectively.

spherical surfaces. We observe the number of molecules that are absorbed/adsorbed to the outer surface or that are inside the inner sphere (as appropriate for each type of molecule).

The average time-varying results from 100 realizations of System 3 are shown in Fig. 17 and compared with the corresponding analytical equations derived in the appendices of [24]⁷. All of the simulations agree very well with their corresponding analytical curves, despite the system not being truly 1D. Thus, we have confidence in AcCoRD’s algorithms for placing molecules within non-trivial spherical environments (since the initialization space was that between the two spheres), detecting imperfect surface reactions, and implementing surface reflections. Videos 4 and 5 in [23] show sample realizations with reversible adsorption and membrane transitions, respectively.

⁷For clarity of exposition and in consideration of space, we do not write out these analytical expressions in this work. However, we note that their general form is similar to the reaction probabilities that we describe in Appendix B.

7.4. System 4: Simple Communication System

In System 4, we consider variations of the extensively-studied molecular communication scenario with a point transmitter and a spherical receiver, as shown in Fig. 4 on page 4. The receiver has a radius of $r = 1 \mu\text{m}$, and the transmitter is placed $d = 5 \mu\text{m}$ from the center of the receiver. Due to the shape of the receiver, and the many variations that we consider, we consistently use a microscopic model. Unless otherwise noted, the transmitter releases an impulse of $N_{\text{TX}} = 10^4$ molecules at time $t = 0$, molecules have a diffusion coefficient of $D = 10^{-10} \frac{\text{m}^2}{\text{s}}$, and every simulation is repeated at least 1000 times.

First, we consider a passive receiver, i.e., the receiver is a passive spherical actor through which the molecules can freely diffuse⁸. In Fig. 18, we test three “outer” boundary conditions, namely: 1) *unbounded*, where molecules are unimpeded everywhere; 2) *reflective*, where the transmitter and receiver are in the middle of a box with a reflective outer boundary; and 3) *absorbing*, where the transmitter and receiver are in the middle of box with an absorbing outer boundary. Each box is a cube and we consider different box widths. The aim of this test is to assess the accuracy of assuming that a bounded environment is unbounded. Fully bounded environments can lead to time-varying impulse responses that include infinite sums; see examples in [7]. However, many envisioned application environments, such as the human body or microfluidic devices, are clearly bounded. We expect that the accuracy of the unbounded model will depend on the time scale considered and (in this case) on the size of the box. The expected time-varying number of molecules at the receiver, $N_{\text{RX}}(t)$, in the unbounded case is a classical result that can be described as a “diffusion wave” and can be expressed from [7, Eq. (3.5)] as

$$N_{\text{RX}}(t) = \frac{N_{\text{TX}}V_{\text{RX}}}{(4\pi Dt)^{\frac{3}{2}}} \exp\left(-\frac{d^2}{4Dt}\right), \quad (26)$$

where V_{RX} is the volume of the receiver, and we assume that the receiver is sufficiently far from the transmitter to assume that the concentration throughout the passive receiver is uniform; see [69].

We plot the average time-varying number of molecules at the receiver for the unbounded and box cases in Fig. 18, where we consider box widths of $\{25, 35, 45\} \mu\text{m}$ and use a time step of $\Delta t = 2 \text{ms}$. The simulation of the unbounded case agrees with the analytical result, and the slight deviation at higher values of t can be improved with more realizations. We expected that the unbounded equation would overestimate the number of molecules inside the absorbing box and underestimate the number of molecules in

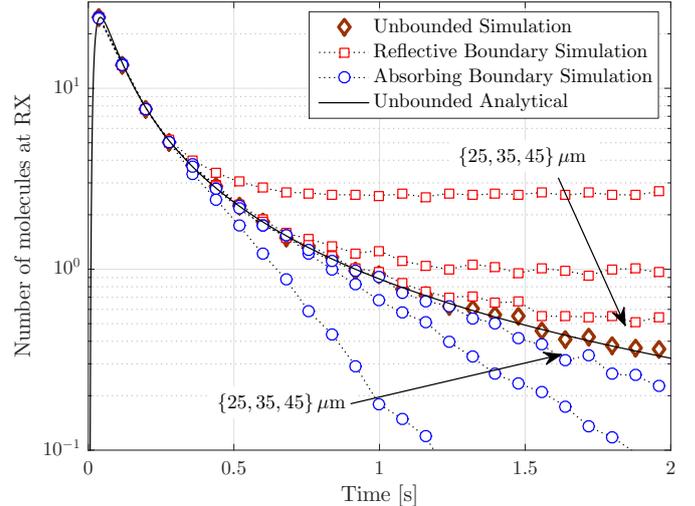


Figure 18: Average time-varying number of molecules inside a passive spherical observer (of radius $1 \mu\text{m}$) after release by a point source located $5 \mu\text{m}$ from the center of the sphere. The unbounded environment is compared with bounded environments that have either a reflective or absorbing outer surface. The bounded environments are cubes with width $\{25, 35, 45\} \mu\text{m}$ with the transmitter and observer in the middle.

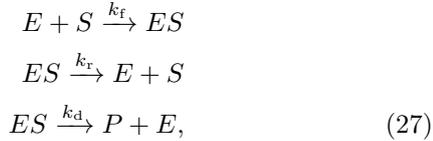
the reflective box. This is indeed the case; the unbounded equation is accurate for early time, and for smaller boxes the deviation from this equation occurs sooner. Interestingly, for the same box size, the reflective and absorbing box models begin to deviate at the same time, i.e., at $\{0.35, 0.75, 1.1\} \text{s}$ when the box is $\{25, 35, 45\} \mu\text{m}$ wide, respectively. In each case, this deviation occurs in about one third of the time it would take for the “peak” of the diffusion wave to reach the edge of the box and return to the source⁹. So, for a bounded environment to accurately model a diffusion-only unbounded environment, the distance from the area of interest to the nearest environment boundary should be at least three times the average 3D diffusion distance of $\sqrt{6Dt}$ for the time scale t of interest.

Next, we study molecule degradation in the propagation environment, which has been considered to address the significant intersymbol interference in a diffusion-only MC system; see [65] and our previous work in [13, 61]. Specifically, we compare molecule degradation through enzymatic action (using Michaelis-Menten kinetics) with passive, first order molecule degradation. With enzyme kinetics, the diffusing information molecules might reversibly bind to diffusing enzyme molecules. Binding to an enzyme also catalyzes the degradation of the information molecule, but this degradation leaves the enzyme intact

⁸We clarify that a “passive” actor is not synonymous with a passive receiver. In AcCoRD, *all* observers are passive actors. For example, a receiver with a reactive surface would still need a passive actor to count the number of molecules on the surface.

⁹It can be shown that the expected time of the peak value of (26) at distance d is $d^2/(6D)$, such that the average diffusion distance for time t is $\sqrt{6Dt}$. For $d = \{25, 35, 45\} \mu\text{m}$, the expected time for molecules to diffuse from the center of the box to the edge and back is $\approx \{1.04, 2.04, 3.38\} \text{s}$, respectively.

to bind with other molecules. The Michaelis-Menten reaction mechanism is commonly used to describe enzyme kinetics and can be written as [11, Sec. 10.2]



where E is the enzyme molecule, S is the substrate molecule (i.e., a molecule released by the transmitter in this context), ES is the intermediate formed by the binding of the enzyme molecule with its substrate, and P is the product molecule (which we will assume is ignored by the receiver, i.e., $P \equiv \emptyset$). k_f , k_r , and k_d are the corresponding reaction rate constants. If the enzyme concentration is uniform, and we assume that both $k_r \rightarrow 0$ and $k_d \rightarrow \infty$, then we can approximate Michaelis-Menten kinetics with a first order mechanism that is



where k_1 is k_f in (27) scaled by the average concentration of enzyme molecules. It is much faster to simulate (28) than (27), due to the extra computational load to compare distances between candidate E and S molecules with the binding radius. Furthermore, analytical solutions for signals with first-order degradation are more readily available. For example, it can be shown that first order degradation changes the diffusion wave in (26) to

$$N_{\text{RX}}(t) = \frac{N_{\text{TX}}V_{\text{RX}}}{(4\pi Dt)^{\frac{3}{2}}} \exp\left(-\frac{d^2}{4Dt} - k_1 t\right), \quad (29)$$

i.e., an additional decaying exponential factor is added.

We compare Michaelis-Menten enzyme kinetics with first order molecule degradation as follows. We seek parameter values that are convenient to simulate. We begin with a target first order degradation rate of $k_1 = 8\text{ s}^{-1}$, a target bimolecular binding radius of $r_B = 0.1\ \mu\text{m}$, and a microscopic time step of $\Delta t = 2\text{ ms}$. The value of k_1 is sufficient to observe measurable degradation over the time scale of the diffusion wave's peak at the passive receiver. The value of r_B is smaller than the size of the average 1D diffusion distance in one microscopic time step $\sqrt{2D\Delta t} \approx 0.63\ \mu\text{m}$, which we need to satisfy in order to relate r_B to k_f with the closed-form expression

$$r_B = \left(\frac{3k_f\Delta t}{4\pi}\right)^{\frac{1}{3}}, \quad (30)$$

as derived in [15, Eq. (27)]. From (30), we calculate that r_B corresponds to $k_f = 2.094 \times 10^{-17}\text{ mol}^{-1}\text{ m}^3\text{ s}^{-1}$, which (fortunately) is almost two orders of magnitude smaller than the largest value of k_f physically possible; see [11, Ch. 10]. The corresponding uniform enzyme concentration needed to achieve an overall binding rate that matches k_1 is $3.82 \times 10^{18}\frac{\text{mol}}{\text{m}^3}$. To limit the actual number of enzyme

molecules needed, we place the transmitter and receiver in the middle of a reflective box of width $15\ \mu\text{m}$. This is significantly smaller than the smallest box considered in Fig. 18, but only requires 12891 enzyme molecules to meet the target concentration. In the interest of computational speed, we proceed with these system parameters, while emphasizing that AcCoRD can also accommodate much larger environments. Most enzymes are proteins that tend to be larger than their substrates (see [4, Ch. 4]), and larger molecules diffuse more slowly, so we set the enzyme E diffusion coefficient to $2 \times 10^{-11}\frac{\text{m}^2}{\text{s}}$ and that of the intermediate ES to $1.8 \times 10^{-11}\frac{\text{m}^2}{\text{s}}$.

In Fig. 19, we observe the average time-varying number of molecules inside the passive receiver when the released S molecules can degrade in the propagation environment, and for additional comparison we include the case where there is no degradation but the transmitter and receiver are still in a reflective box of width $15\ \mu\text{m}$. We see that the simulation of first order degradation agrees well with the analytical expression in (29) and shows significant deviation from the signal without degradation given by (26) and the simulation of the bounded environment. Interestingly, the curve for the simulation of the bounded environment appears to reach a minimum after about 0.25 s and then increase. This is because molecules are returning to the center of the box after reflecting off of the boundary. The receiver occupies about 1.24% of the box, so the expected number of molecules should converge to about 12.4.

For the enzyme kinetics in Fig. 19, we simulate three variations in the values of k_r and k_d , i.e., in the reverse (unbinding) and degradation reactions, respectively. In the ideal limit, where $k_r \rightarrow 0$ and $k_d \rightarrow \infty$, the first order degradation curve is still a clear lower bound. Near the peak of the curve, this difference is because a given enzyme molecule can only react with one observable molecule in a given time slot. For time $t > 0.1\text{ s}$, the finite size of the environment also becomes a factor, although not to the degree observed in the absence of degradation. We also consider very slow but practical values of k_r and k_d ¹⁰, by first setting $k_d = 1\text{ s}^{-1}$ and then adding $k_r = 1\text{ s}^{-1}$. When k_d is finite, an intermediate molecule can exist for multiple time slots. This slows down an enzyme molecule's capacity to bind with multiple substrates, so overall fewer molecules degrade and more are observed at the receiver. When we include a non-zero value of k_r , intermediates can also unbind with no change to the substrate, which further reduces the amount of degradation. From Fig. 19, we clearly see that the first order degradation model has limited accuracy in modeling practical enzyme kinetics. Video 6 in [23] shows a sample realization with non-zero k_r and k_d .

For our final variation of a passive receiver, we study the time-varying statistics of the receiver's observations

¹⁰ k_d and k_r typically have values between 1 and 10^5 s^{-1} ; see [11, Ch. 10].

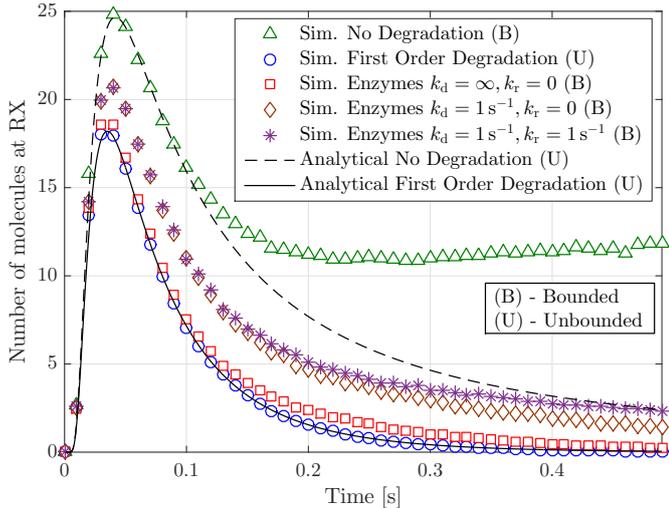


Figure 19: Average time-varying number of molecules inside a passive spherical observer (of radius $1 \mu\text{m}$) after release by a point source located $5 \mu\text{m}$ from the center of the sphere. An unbounded environment where the molecules undergo first order degradation is compared with 1) a bounded, reflective environment ($15 \mu\text{m}$ in width) where the molecules can degrade by binding with diffusing enzymes, and 2) the same reflective environment without degradation. Analytical curves are included for first order degradation and no degradation in an unbounded environment.

when the transmitter releases a series of molecule pulses with finite width. Specifically, the transmitter encodes the 10-bit sequence $[1, 1, 0, 1, 0, 1, 1, 1, 0, 1]$ with a bit interval of 0.1s . For every bit 0, no molecules are released. For every bit 1, molecules are released as a zeroth order reaction for 0.02s at a rate of $5 \times 10^5 \text{ mol/s}$, i.e., on average there are 10^4 molecules released for every bit 1. The released molecules also degrade at a rate of $k = 8\text{s}^{-1}$. The time-varying PMF for the observations at the receiver is plotted in Fig. 20. We can distinguish the seven peaks for the bit 1s, but also notice the accumulation of intersymbol interference since repeated peaks get higher (and this would be even more noticeable in the absence of molecule degradation). The information used to generate Fig. 20 can also be used to analyze the detection performance at the receiver, but that is not a feature currently native to AcCoRD and so is outside the scope of this paper. Video 7 in [23] shows the first five symbol intervals of a sample realization of this multi-symbol variation.

Finally, we consider variations of System 4 with reactions at the surface of the receiver. Such a receiver more accurately models the chemical detection of molecules. We focus on the fully-absorbing (i.e., perfectly-absorbing) receiver and the partially-absorbing receiver, which have well known analytical results for the geometry of System 4. When the receiver is fully absorbing, the expected time-varying number of molecules absorbed is given by [70,

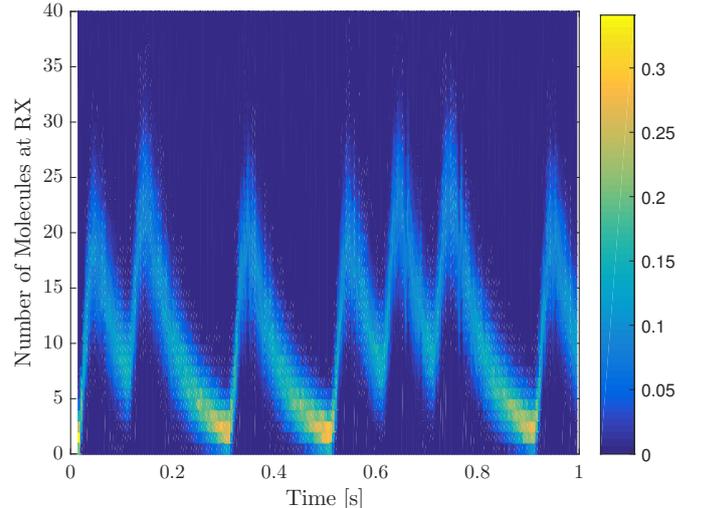


Figure 20: Time-varying probability mass function of the number of molecules inside a passive spherical observer when a point source releases rectangular pulses of molecules according to the binary sequence $[1, 1, 0, 1, 0, 1, 1, 1, 0, 1]$ with a symbol interval of 0.1s . The pulses have a length of 0.02s and a strength of $5 \times 10^5 \text{ mol/s}$. Released molecules degrade at rate $k = 8\text{s}^{-1}$. The bar to the right is the legend for observation probabilities.

Eq. (3.116)]

$$N_{\text{RX}}(t) = \frac{N_{\text{TX}}r}{d} \operatorname{erfc} \left(\frac{d-r}{\sqrt{4Dt}} \right). \quad (31)$$

When the receiver is partially absorbing, the absorption rate k is finite. The expected time-varying number of molecules absorbed is given by [70, Eq. (3.114)]

$$N_{\text{RX}}(t) = N_{\text{TX}} \frac{r\beta - 1}{d\beta} \left[1 + \operatorname{erf} \left(\frac{r-d}{2\sqrt{Dt}} \right) - \exp \left((d-r)\beta + Dt\beta^2 \right) \operatorname{erfc} \left(\frac{d-r+2D\beta t}{2\sqrt{Dt}} \right) \right], \quad (32)$$

where

$$\beta = \frac{k}{D} + \frac{1}{r}. \quad (33)$$

We plot the average time-varying number of molecules absorbed by the receiver in Fig. 21, where we consider full absorption and partial absorption with a rate of $k = 200 \mu\text{m/s}$. A time step of $\Delta t = 1 \mu\text{s}$ was used to accurately simulate the receiver reactions. Both simulations agree very well with their corresponding analytical curves. Video 8 in [23] shows a sample realization of the partially-absorbing receiver.

8. Limitations and Future Development

As we have previously noted, the development of AcCoRD is active and ongoing. In this section, we discuss

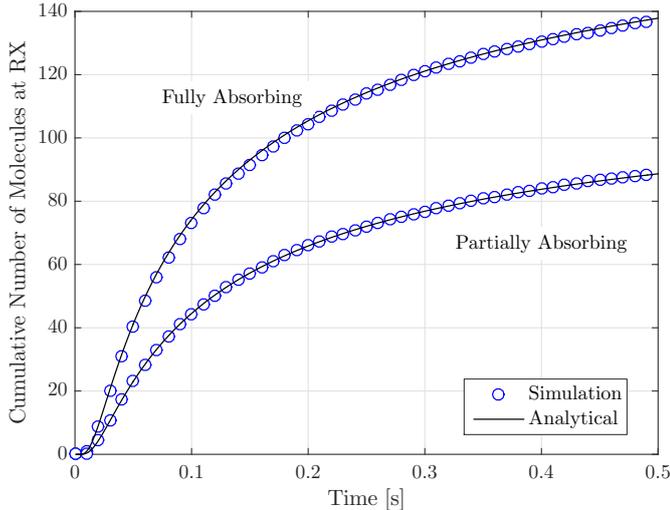


Figure 21: Average time-varying number of molecules absorbed by a spherical surface (of radius $1\ \mu\text{m}$) after release by a point source located $5\ \mu\text{m}$ from the center of the sphere. Infinite ($k = \infty$) and finite ($k = 200\ \mu\text{m}/\text{s}$) absorption rates are considered for the fully and partially absorbing cases, respectively.

some of AcCoRD’s current limitations and how they may be addressed in the future to improve either the accuracy or the applicability of the software. We focus on high level limitations that apply to the implementation of reaction-diffusion phenomena or the design of molecular communication systems. A more detailed description of technical constraints is included in Appendix A.

8.1. Adding Molecule Flow and Polarity

We have limited molecule behavior to pure diffusion and selected chemical reactions. We can potentially make minor improvements to the implementations of these phenomena, such as by accounting for environment boundaries in the hybrid transition rules and for surface reactions occurring within a microscopic time step. Furthermore, there are other phenomena that can influence a molecule’s behavior. Molecules could be carried by a bulk fluid flow that can vary in both space and time. For example, laminar flow, where layers of fluid slide over one another, is predominant in small blood vessels; see [71]. Also, molecular interactions could be introduced by the presence of ionic charges, such that the molecules exert attractive or repulsive forces on each other. These interactions were first considered in a molecular communication context in [72].

8.2. Expanding Actor Placement and Behavior

Actors in AcCoRD must currently be fixed in place and they have independent, predetermined behavior. This is sufficient for unidirectional communication between two fixed devices, but cannot model more complex systems that may require bidirectional communication, relaying,

mobility, or other behaviors that rely on real-time signal processing. In both practical biological environments and communication networks, the detection of a signal should trigger a response at the receiver. Under the framework of AcCoRD, we should couple active actors to passive actors so that molecules can be released when specified observation thresholds are met (i.e., detectors are needed to trigger the active actor). We can also consider introducing actor mobility, which can be more accurate for modeling transceivers that are able to move (e.g., bacteria).

Active actors can only release molecules according to the modulation of a concentration shift keying signal, where the number (or rate) of molecules released is linearly proportional to the current information symbol. Other types of modulation can be added, such as those described in [53]. Furthermore, adaptive transmission schemes can be considered, where the number of molecules released for a given symbol depends on the interference expected from previous symbols, as we considered in [73].

8.3. Improving Scalability

In the simulations of System 2, we observed conditions where the hybrid simulation demonstrated considerable improvement in computation speed over the microscopic model. The potential trade offs in simulation accuracy versus computational complexity can be further improved by adding “tau-leaping” to the mesoscopic model; see [34, 35, 74]. In tau-leaping, mesoscopic time steps are used to execute multiple reaction and diffusion events simultaneously. If mesoscopic subvolumes have a sufficient number of molecules, then tau-leaping can improve the simulation run time considerably with minimal loss of accuracy. Finally, a possibility for scalability in the microscopic regime is to enable a different time step in each region.

9. Conclusions

In this paper, we introduced the AcCoRD simulator (Actor-based Communication via Reaction-Diffusion). AcCoRD is a generic 3D reaction-diffusion solver that is designed for MC analysis. It can use a combination of microscopic and mesoscopic simulation models to track time-varying molecule behavior over independent realizations. Simulation environments are defined by the flexible placement of regions. Actors are used to release or observe molecules in the regions. Utilities are provided to visualize the simulation environment and to plot the time-varying behavior or observation statistics. We described all of the simulation components, including the underlying reaction-diffusion theory. We discussed the overall AcCoRD algorithm, including its run time complexity, and we also described the work flow for running and processing a simulation. We demonstrated AcCoRD’s consistency with analytical results via a detailed series of simulations.

We hope that the development of AcCoRD will encourage a wider use of simulations within the MC community

to support analytical modeling and predict experimental results. MC is a multi-disciplinary field and the availability of a relevant reaction-diffusion “sandbox” should improve its accessibility to new researchers while providing a flexible yet convenient platform for exploration and verification.

Acknowledgments

This research was funded in part by the Natural Sciences and Engineering Research Council of Canada via a Postdoctoral Fellowship, by the University of British Columbia via a Four-Year-Fellowship, and by the Université de Montréal. Computing resources were provided by the Centre for Advanced Computing and Compute/Calcul Canada. The authors would like to thank Modar Halimeh and Tobias Schwering for their help and feedback with software testing.

Appendix A. Constraints and Limitations

In this appendix, we provide additional detailed notes on AcCoRD’s current technical constraints and limitations.

The following constraints exist for the placement of regions:

- Two regions can overlap only if one region is entirely inside the other. In that case the inner (i.e., *child*) region must identify the outer region as its *parent* (to let AcCoRD know that the nesting is intended). Multiple levels of nesting can be used, i.e., a child’s parent can also have its own parent region.
- If a spherical region (which must always be microscopic) is the parent of a box region (or vice versa), then their surfaces must be separated by at least a distance of the box region’s subvolume length h . If the box region is the parent of the spherical region, then the box must also be microscopic. However, a spherical region can have a mesoscopic child region.
- If a box region is the parent of a box region, then the placement of the child must be flush with subvolumes of the parent (i.e., every subvolume in the parent region is either fully within the child or not). This applies whether the parent is microscopic or mesoscopic.
- Regions can be placed adjacent to each other, including as children of the same parent region.
- A square surface cannot have a normal 3D region as its parent, but they can be placed adjacent to each other.
- Surface regions must be microscopic and can only be adjacent to other microscopic regions. This could be relaxed in a future release.

- Regions that are supposed to be neighbors but which are separated by multiple levels of nesting (e.g., grandparents and grandchildren regions) are not properly detected as neighbors. This occurs when a region shares part of its outer boundary with its parent but the parent does not share part of its outer boundary with the grandparent. This could be corrected in a future release.

The following constraints exist for the placement of actors:

- Spherical actors must be placed entirely in the microscopic regime.
- If two actors are configured to act at the same time, then they will act in a random order (according to whichever actor happens to be at the top of the timer queue). For example, if a user intends to observe molecules immediately after they are placed, then they should add a small offset to the start time of the passive actor. This could be improved in a future release by assigning an actor priority to define which actor should go “first” in the case of simultaneous behavior.

The following constraints and limitations exist for chemical reactions:

- Membrane surface reactions can only be defined at membrane surface regions.
- Absorbing and desorbing surface reactions can only be defined at non-membrane surface regions.
- There is currently no mechanism to enable a surface-bound molecule to diffuse along its surface, which should be possible if the surface is fluid (e.g., lipid bilayers in cells are 2D fluids whose molecules can move in the plane of the bilayer; see [4, Ch. 11]).
- Surface reactions only occur by testing a candidate molecule’s diffusion trajectory for crossing a reactive surface. It is more accurate to also consider the possibility that a molecule reacted with a surface even if its diffusion trajectory does not intersect the surface. We ignored this more general case, as in [24], but this might be added in a future release.
- Unbinding radii are only applied to *second order* microscopic reactions that have multiple products. This could be expanded in a future release to apply to all microscopic reactions with multiple products, including the unbinding of enzyme intermediates.
- Bimolecular reactions in the microscopic regime are tested at the *region* level, which can result in many unnecessary distance comparisons if the region is much larger than the binding radius. One workaround is to define regions on the order of the

size of the binding radius. A future release could instead test bimolecular reactions at the level of microscopic *subvolumes*, although this would introduce overhead to keep track of which subvolume every candidate reactant is in.

The following limitations exist for hybrid diffusion between mesoscopic and microscopic regions:

- The assumption on subvolume size at a hybrid interface (i.e., assuming that subvolume length h is $\gg \sqrt{D\Delta t}$ or $\sim \sqrt{D\Delta t}$) is applied uniformly to the entire environment and does not consider local values of diffusion coefficient or subvolume size. A future release could enable local definition of this assumption based on the specific interface and diffusing molecule.
- Mesoscopic molecules added to the microscopic regime via transition rule (9), i.e., under the assumption that the subvolume is small, could be placed beyond the environment boundary. The validity of the molecule’s location is not corrected until the following microscopic time step. This can be corrected in a future release.
- A mesoscopic subvolume that is adjacent to a microscopic region along some face of the subvolume should be “exposed” to that region along the *entire* face. The equation for the diffusion propensity in (5) assumes that this is the case.

Appendix B. Surface Reaction Probabilities

In this appendix, we list and describe the surface reaction probabilities. We also discuss the placement of the corresponding product molecules. Most equations in this appendix were derived in [24] and are included here because they are implemented directly in AcCoRD. All probabilities are described for arbitrary reaction c with reactant m and reaction rate¹¹ constant k_c . Where applicable, k_{c^*} is the reaction rate of reaction c ’s reverse reaction. A direct comparison of the accuracy of the different equations for surface reactions is outside the scope of this work.

Generally, [24] derives “steady state” reaction probabilities which imply that the interactions between the molecules and the surface are in a state of equilibrium. However, these probabilities were shown to also be accurate over transient time scales. In some cases there is also a “well-mixed” reaction probability which assumes that the molecule distribution in the vicinity of the surface is homogeneous. The well-mixed probability, originally derived in [75], is in general not as accurate as the steady state expressions but it is simpler to calculate.

¹¹Strictly speaking, absorption and membrane reactions are defined by a reaction *coefficient*, but we use the term “rate” here for consistency.

Appendix B.1. Absorption

First, we consider absorption, where a molecule binds to or is consumed by the surface. If absorption occurs, then the product molecule(s) is (are) placed at the surface where the surface intersected the trajectory of the diffusing reactant. If the reaction is unsuccessful, then the molecule has a perfectly-elastic collision with the surface and is reflected off of the surface at the intersection point.

The well-mixed absorption probability is [24, Eq. (1)]

$$P_{\text{abs}}|_{\text{mixed}} = k_c \sqrt{\frac{\pi \delta t}{D_m}}, \quad (\text{B.1})$$

where δt is the corresponding time step (which can be less than or equal to the simulation’s microscopic time step Δt , depending on when the molecule was created), and D_m is the molecule’s diffusion coefficient. Eq. (B.1) is accurate if

$$k_c \ll \sqrt{\frac{D_m}{\pi \delta t}}, \quad (\text{B.2})$$

i.e., if P_{abs} is sufficiently small.

The steady state reaction probability depends on whether the absorption is reversible. In the irreversible case, a polynomial fit was used to find the reaction probability as [24, Eq. (21)]

$$P_{\text{abs,irr}}|_{\text{ss}} = k'_c \sqrt{2\pi} - 3.33321k'_c{}^2 + 3.35669k'_c{}^3 - 1.52092k'_c{}^4, \quad (\text{B.3})$$

where [24, Eq. (12)]

$$k'_c = k_c \sqrt{\frac{\delta t}{2D_m}}, \quad (\text{B.4})$$

is the reduced absorption coefficient.

The steady state reaction probability for reversible absorption (i.e., *adsorption*) was derived in closed form as [24, Eq. (37)]

$$P_{\text{abs,rev}}|_{\text{ss}} = \frac{k'_c \sqrt{2\pi} (\gamma_2 - \gamma_1 - \gamma_2 \operatorname{erfcx}(\gamma_1) + \gamma_1 \operatorname{erfcx}(\gamma_2))}{\gamma_1 \gamma_2 (\gamma_2 - \gamma_1)}, \quad (\text{B.5})$$

where

$$\gamma_1 = \frac{k'_c - \sqrt{k'_c{}^2 - 2k_{c^*} \delta t}}{\sqrt{2}}, \quad (\text{B.6})$$

$$\gamma_2 = \frac{k'_c + \sqrt{k'_c{}^2 - 2k_{c^*} \delta t}}{\sqrt{2}}, \quad (\text{B.7})$$

k_{c^*} is the rate of the reverse desorption reaction, $\operatorname{erfcx}(\cdot)$ is the scaled complementary error function

$$\operatorname{erfcx}(\gamma) = \exp(\gamma^2) (1 - \operatorname{erf}(\gamma)), \quad (\text{B.8})$$

and the error function $\operatorname{erf}(\cdot)$ is defined in (7).

Appendix B.2. Desorption

Desorption releases a molecule that was bound to the surface. In the irreversible case, the probability of reaction is the same as that for a non-surface first order reaction, i.e., as in (16). However, we must still determine where to place the product. We place the product along the surface normal from the point where the reactant was bound. A steady state distance x along the normal can be generated via [24, Eq. (27)]

$$x = \frac{0.571825u - 0.552246u^2}{1 - 1.53908u + 0.546424u^2} \sqrt{2D_m \delta t}, \quad (\text{B.9})$$

where D_m is the diffusion coefficient of the molecule once it is unbound from the surface and u is a single uniform random variable between 0 and 1. Eq. (B.9) was determined numerically with the assumption that the precise reaction time was unknown. In AcCoRD, we can determine the time of irreversible desorption using the method described in Section 4.3. Given desorption time t_1 , the molecule will have $\delta t - t_1$ seconds to diffuse from the surface, and the distance x along the normal can be generated via

$$x = \sqrt{8D_m(\delta t - t_1)}n, \quad (\text{B.10})$$

where n is a normal random variable with mean 0 and variance 1. Eq. (B.10) is twice the unbounded diffusion distance for this time interval because here we have a reflective surface (since the desorption is irreversible), and in AcCoRD we use (B.10) by specifying the “full diffusion” option for placing desorbed molecules.

The reaction probability for reversible desorption can be written as a function of the corresponding absorption (i.e., adsorption) reaction in (B.5), i.e., [24, Eq. (32)]

$$P_{\text{des,rev}}|_{\text{ss}} = \frac{k_c}{k_{c^*}} \sqrt{\frac{D_m \delta t}{\pi}} P_{\text{abs,rev}}|_{\text{ss}}, \quad (\text{B.11})$$

where here k_c is the desorption rate and k_{c^*} is the rate of the reverse adsorbing reaction.

Since the exact desorption times are unknown in the reversible case, we should only consider molecule placement via the corresponding steady state equation, which was determined numerically as [24, Eq. (35)]

$$x = \frac{0.729614u - 0.70252u^2}{1 - 1.47494u + 0.484371u^2} \sqrt{2D_m \delta t}. \quad (\text{B.12})$$

Appendix B.3. Membrane Transitions

Membrane reactions enable selective transitions across a surface. When these reactions occur successfully, the diffusing molecule continues along its original trajectory. Otherwise, it is reflected off of the surface at the point where the trajectory intersected the surface.

In the irreversible case, a molecule can only transition in one direction. Thus, the transition probabilities in this case are the same as for an absorption reaction with the

same reaction rate, i.e., the well-mixed probability is given by (B.1) and the steady state probability is given by (B.3).

The forward and reverse reactions in reversible membrane transitions are analogous to each other. Thus, unlike the irreversible case, reversible membrane reactions are not analogous to reversible adsorption. The transition probability for reversible membrane transitions was derived as [24, Eq. (47)]

$$P_{\text{mem,rev}}|_{\text{ss}} = \frac{k_c}{(k_c + k_{c^*})^2} \left[2(k_c + k_{c^*}) + \sqrt{\frac{\pi}{2}} \left(\text{erfcx} \left(\sqrt{2}(k_c + k_{c^*}) \right) - 1 \right) \right], \quad (\text{B.13})$$

where k_c is the reaction rate for the transition in the direction being considered.

Appendix C. Detailed AcCoRD Algorithms

In this appendix, we describe each of the four system evolution steps of the overall AcCoRD algorithm that was presented in Section 5.1. Specifically, we describe active actor behavior, passive actor behavior, evolution of the microscopic regime, and evolution of the mesoscopic regime. We also include comments as appropriate on the computational speed of the microscopic and mesoscopic algorithms using big O notation ($\mathcal{O}(\cdot)$).

Appendix C.1. Active Actors

The algorithm for executing an active actor simulation step is presented in Algorithm 2. Each step takes one of two actions: either a new symbol is *created* for some actor a , or molecules from an existing symbol are released by actor a . We separate these two actions in order to accommodate the release of molecules over a *release interval*. The release interval can be instantaneous or any finite interval independent of the actor’s *action interval*, i.e., symbol interval. By defining the release interval to be longer than the action interval, it is technically possible for an actor to be releasing molecules for multiple symbols simultaneously (although this is most likely impractical for a molecular communication system).

When a new symbol is needed, we generate the symbol bits randomly (with some independent probability) or read them from a sequence provided by the configuration file. The modulation properties and the new symbol bits are used to determine how many molecules to release and how often. For example, concentration shift keying (CSK) releases molecules at a uniform rate over the release interval, where the rate is linearly proportional to the symbol value.

When the current actor action is the actual release of molecules, we use the modulation properties to determine how many molecules to release at the current time. If the release times within the interval are random, then only one

Algorithm 2 Active Actor Step

```
1: procedure ACTIVE_ACTION( $a$ )
2:   if Next actor  $a$  action is new symbol then
3:     if Bits are random then
4:       Generate random symbol bits
5:     else
6:       Read defined symbol bits
7:     end if
8:     Determine modulation parameters
9:     Update list of current symbols
10:    Update time of next new symbol
11:  else  $\triangleright$  Releasing molecule(s) for a current symbol
12:    if Release times random then
13:       $N_{\text{new}} \leftarrow 1$   $\triangleright$  Only one new molecule
14:      Generate next release time
15:    else
16:      Calculate  $N_{\text{new}}$ 
17:      Calculate next release time
18:    end if
19:    for All  $N_{\text{new}}$  do
20:      Place new molecule within actor  $a$ 
21:    end for
22:    Update time of symbol's next molecule release
23:  end if
24:  Determine actor  $a$ 's next action
25:  Update actor  $a$ 's timer
26: end procedure
```

molecule will be released and we must randomly generate the next molecule release time as a zeroth order reaction process (see Section 4.3 for the simulation of zeroth order reactions). Otherwise, the number of molecules will depend on the modulation strength, and we calculate the next molecule release time using a *slot interval*, which is a subset of the release interval. In either case, if the next molecule release time is beyond the current release interval, then it is ignored and transmission for the corresponding symbol is complete.

Every molecule released by actor a is initialized at a uniformly-generated random location within the actor volume. If the location is within the microscopic regime, then the molecule is added to the corresponding region's linked list for that molecule type. If the location is within the mesoscopic regime, then the molecule count in the corresponding subvolume is incremented and the subvolume's reaction propensities are updated as described in Appendix C.4.

At the end of the active actor simulation step, we determine actor a 's next action by comparing the time of the next new symbol with that of the next molecule release. The corresponding time is used to update the actor's timer.

Algorithm 3 Passive Actor Step

```
1: procedure PASSIVE_ACTION( $a$ )
2:   for Each type of molecule being observed do
3:     for Every region within actor  $a$  do
4:       if Region microscopic then
5:         for Every molecule in "recent" list do
6:           if Molecule within actor  $a$  then
7:             Record molecule
8:           end if
9:         end for
10:        Repeat lines 5 to 9 for "normal" list
11:       else  $\triangleright$  Region mesoscopic
12:         for Every subvolume within actor  $a$  do
13:           if Entire subvolume inside actor
14:             then
15:               Record all molecules
16:             else  $\triangleright$  Partial overlap
17:               for Every molecule do
18:                 Flip biased coin for location
19:                 if Molecule within actor  $a$ 
20:                   then
21:                     Record molecule
22:                   end if
23:                 end for
24:               end if  $\triangleright$  Subvolumes in region
25:               end if  $\triangleright$  Microscopic or mesoscopic?
26:             end for  $\triangleright$  Regions in actor  $a$ 
27:              $\triangleright$  Molecule types
28:           Append observations to realization list
29:           Update actor  $a$ 's timer
30:         end for
31:       end for
32:   end procedure
```

Appendix C.2. Passive Actors

The algorithm for executing a passive actor simulation step is presented in Algorithm 3 for passive actor a . We test the location of all candidate molecules against the boundary of the actor. This test is trivial for molecules whose region is known to be entirely inside the actor. Any molecule that is within the actor is recorded. In general, we only count the *number* of each type of molecule that the actor is configured to observe. Optionally, the observation can also include the molecule location coordinates. These coordinates are necessary if making a video from the simulation output.

In microscopic regions, the precise coordinates of every molecule are already known and so the test for being inside actor a is a purely geometric problem. It is also straightforward to record molecule locations if needed. In mesoscopic regions, we consider the individual subvolumes. If a subvolume is known to be inside the actor, then all corresponding molecules are recorded. Otherwise, we flip a biased coin for each molecule based on the relative fraction of the subvolume that is within actor a . If molecule locations are needed, then they are randomly generated

within the subvolume (or a fraction thereof when the entire subvolume is not within the actor).

Once all molecules have been recorded, we append this information to the master list of observations for the current realization. Finally, we update actor a 's timer by incrementing it by the actor's *action interval*.

Appendix C.3. Microscopic Steps

The algorithm for executing a microscopic simulation step is presented in Algorithm 4. The primary stages in this algorithm are (in order) the execution of zeroth order reactions, first order reactions, diffusion, and second order reactions. For clarity of presentation in Algorithm 4, we omit showing that each stage is completed for all regions before proceeding to the next stage.

Appendix C.3.1. Zeroth Order Reactions

Zeroth order reactions in a microscopic region each have their own internal timer to track the time of the most recent reaction event. Within a microscopic time step, we generate the delay between reaction times as exponential random variables according to (12) in Section 4.3. We continue until the reaction time exceeds the time remaining in the step and is saved for the next time step. Random reaction locations are generated uniformly within the region volume. Every product molecule is added to the region's corresponding "recent" molecule list with the reaction location and the molecule's time step.

The generation of zeroth order reaction times is effectively analogous to how zeroth order reaction events can occur in the mesoscopic regime. Thus, this process for zeroth order reaction c in region r takes $\mathcal{O}(k_c V_r)$ time, where V_r is the region's volume.

Appendix C.3.2. First Order Reactions

The stage for *non-surface* first order reactions is as follows. For each molecule that can be the reactant in a first order reaction, we generate a uniform random variable u between 0 and 1 and compare it with the probability of the reaction occurring in (16) (or with the probabilities generated by (17) if the molecule is a potential reactant in more than one non-surface first order reaction). If a reaction is successful, then we generate the precise reaction time as a constrained exponential random variable via (18). Product molecules are initialized at the same location as the reactant. This process is iterative, so a product molecule that can also be the reactant in a first order reaction is tested for reacting in the time still remaining within the time step. As with zeroth order reactions, the generation of reaction times is effectively analogous to how it is done in the mesoscopic regime, so the number of instances of first order reaction c grows with $\mathcal{O}(k_c N_{c,r})$, where $N_{c,r}$ is the number of corresponding reactant molecules in the region.

Algorithm 4 Microscopic Regime Step

```

1: procedure MICROSCOPIC_STEP
2:   for Every zeroth order reaction type do
3:     while Next reaction time within time step do
4:       Execute reaction
5:       Generate next reaction time
6:     end while
7:   end for ▷ Zeroth order reactions
8:   for Every molecule do
9:     while First order reaction is possible do
10:      Flip coin for first order reaction
11:      if Reaction occurs then
12:        Generate reaction time
13:        Execute reaction
14:      end if
15:    end while
16:   end for ▷ First order reactions
17:   for Every molecule do
18:     Generate new location via diffusion
19:     Validate new location
20:     if Molecule entered mesoscopic regime then
21:       Place molecule in subvolume
22:     else if Molecule reacted with surface then
23:       Execute reaction
24:     else
25:       Update molecule in "normal" list
26:     end if
27:   end for ▷ Diffusion
28:   for Every second order reaction do
29:     for Every valid first reactant do
30:       for Every valid second reactant do
31:         if Reactants within binding radius then
32:           if Reactants reach reaction site then
33:             Execute reaction
34:             Separate with unbinding radius
35:           end if
36:         end if
37:       end for ▷ Second reactants
38:     end for ▷ First reactants
39:   end for ▷ Second order reactions
40:   Update subvolume propensities as needed
41:   Increment microscopic timer by  $\Delta t$ 
42: end procedure

```

Appendix C.3.3. Diffusion

The diffusion stage generates a potential diffusion trajectory for every molecule with a nonzero diffusion coefficient. The time to generate and test the trajectories grows with $\mathcal{O}(N_{\text{micro},r})$, where $N_{\text{micro},r}$ is the number of molecules in microscopic region r . We follow each proposed trajectory from the initial location to the proposed new location in a process that we approximate as taking $\mathcal{O}(|\Omega_{\text{neigh},r}|)$ time, where $|\Omega_{\text{neigh},r}|$ is the number of regions that are neighbors of region r . In practice, $\mathcal{O}(|\Omega_{\text{neigh},r}|)$ time can be considerable, since we need to

check every neighbor for a trajectory intersection and implement the corresponding behavior at the closest intersecting neighbor. This process must then be repeated for every boundary reflection or transition to a normal (i.e., non-surface) neighbor.

Molecules are permitted to transition between adjacent normal regions without impedance. If the trajectory crosses a region boundary where there is no adjacent region, then we reflect the trajectory from the point of intersection with the boundary. If the trajectory crosses a surface region, then we test the corresponding surface reaction (either absorption or a membrane transition, if applicable) with its probability calculated in Appendix B and we reflect the trajectory if the surface reaction fails. If absorption occurs, then we execute the corresponding reaction with a reaction time that is at the *end* of the time step. If a membrane transition occurs, then the molecule can pass through the surface and we continue to test the proposed trajectory.

If the diffusion trajectory crosses a mesoscopic region, then we place the molecule in the first mesoscopic subvolume crossed by the trajectory. Even if a molecule’s trajectory does not cross a mesoscopic region, it can be placed in a mesoscopic region if the molecule’s final location is in a region that neighbors a mesoscopic region, which we test by comparing a uniform random variable u with the probability in (4). This probability is a function of the initial and final distances from the hybrid interface. Performing this test takes considerable time, since we need to calculate the corresponding distances and generate a random variable, so we ignore this test if either the initial or final distance is greater than the defined maximum distance l_{\max} . Every molecule that is still in the microscopic regime at the end of the diffusion stage is in the corresponding region’s “normal” list.

Appendix C.3.4. Second Order Reactions

In the stage for second order reactions, we compare the distances between pairs of candidate reactants for every possible reaction. This takes up to $\mathcal{O}\left(N_{\text{micro},r}\left(N_{\text{micro},r} + \sum_{\Omega_{\text{neigh},r}} N_{\text{micro},w}\right)\right)$ time for each reaction in region r , where the summation is over the set of regions that are neighbors to region r and $N_{\text{micro},w}$ is the number of microscopic molecules in neighboring region w . To be a candidate pair, the two molecules must satisfy all of the following criteria:

- Be the specific reactants for the current second order reaction.
- Be located in the same or neighboring microscopic regions.
- Have not yet participated in a second order reaction within the current time step (either as a reactant or product).

If a candidate pair of reactants are separated by less than the reaction binding radius r_B , then the candidate reaction site is determined from the reactants’ diffusion coefficients D_A and D_B . If the current locations (i.e., after diffusing) of reactants 1 and 2 are $\vec{r}_1 = \{x_1, y_1, z_1\}$ and $\vec{r}_2 = \{x_2, y_2, z_2\}$, respectively, then the candidate reaction site \vec{r}_{rxn} is

$$\vec{r}_{\text{rxn}} = \frac{D_1}{D_1 + D_2} (\vec{r}_2 - \vec{r}_1) + \vec{r}_1. \quad (\text{C.1})$$

To validate \vec{r}_{rxn} , we follow the trajectory of each reactant from their current location. Both reactants must be able to reach \vec{r}_{rxn} . We deem \vec{r}_{rxn} to be invalid if *any* of the following are true:

- Either reactant reflects off of a surface.
- Either reactant crosses a mesoscopic region.
- \vec{r}_{rxn} is in a region where the current second order reaction was not defined.

We assume that all potential surface reactions at this stage would lead to reflection (which invalidates the candidate second order reaction site) because we force the time step for reactions to $\delta t = 0$. This is an approximation since we assume that the translation of reactants to the reaction site occurs spontaneously at the end of the current time step (since surface reactions were already assessed for each reactant in the diffusion stage). If \vec{r}_{rxn} is valid, then we remove the reactants and add product molecules that are centered around \vec{r}_{rxn} according to the unbinding radius r_U . Otherwise, the reaction does not occur and the reactants remain at their current locations. If there is one product, then it is placed at \vec{r}_{rxn} . If there are at least two products, then the displacement of each product is r_U scaled by the relative diffusion coefficient (similar to how we determined \vec{r}_{rxn} in (C.1)). If there are two products, then they are placed in the same directions as the locations of the two reactants. If there are more than two products, then the directions are determined randomly. We track the trajectory of each product molecule from the reaction site in case the molecule reflects off of a surface or enters a mesoscopic region, and correct the product molecule’s location as needed.

At the end of the microscopic simulation step, we update the propensities of mesoscopic subvolumes that had molecules added via diffusion (as described in Appendix C.4), and then we increment the microscopic timer by the microscopic time step Δt .

Appendix C.4. Mesoscopic Steps

The algorithm for executing a mesoscopic simulation step is presented in Algorithm 5. Each potential reaction event in every mesoscopic subvolume, including chemical reactions and diffusion between adjacent subvolumes, is assigned a propensity, α . We described the calculation of event propensities in Section 4 as functions of the type of

Algorithm 5 Mesoscopic Regime Step

```
1: procedure MESOSCOPIC_STEP
2:   Determine subvolume of next reaction (via NRM)
3:   Determine next reaction in subvolume (via DM)
4:   if Next event is diffusion then
5:     Determine molecule type and destination
6:     Decrement molecule count in source subvolume
7:     if Destination subvolume is microscopic then
8:       Create microscopic molecule
9:     else           ▷ Destination is also mesoscopic
10:      Increment molecule count in destination
11:      Update destination propensity (via NRM)
12:     end if
13:   else           ▷ Next event is chemical reaction
14:     Adjust molecule counts
15:   end if
16:   Update source propensity (via NRM)
17:   Update mesoscopic timer
18: end procedure
```

event, the subvolume size, and the number of each type of molecule in the subvolume.

The primary feature in a mesoscopic algorithm is how the reaction propensities are used to determine the order of reaction events, when the events occur, and how the order is updated. There are several mathematically equivalent methods for doing so, and they can be integrated as desired for a given implementation. In AcCoRD we adopt the Next Subvolume Method (NSM), which was originally proposed in [76] and uses both Gillespie’s original Direct Method (DM; see [46]) and Gibson and Bruck’s Next Reaction Method (NRM; see [77]). Specifically, we use the NRM to determine the subvolume where the next event will occur and the DM to determine the next event in a given subvolume. Even though the NRM is efficient when there is a very large number of events, the NSM was shown in [76] to be faster than applying the NRM to all events.

The primary difference between the original NSM and the implementation in AcCoRD is that AcCoRD accounts for the hybrid interface between microscopic and mesoscopic regimes. The event propensity for a subvolume molecule to enter a microscopic region, and the algorithm for determining a new microscopic molecule’s location, were described in Section 4.2.2.

Appendix C.4.1. Next Reaction Method in NSM

The NSM uses the NRM to determine the subvolume where the next reaction occurs. We sort all subvolumes in an indexed priority queue. Subvolume s has an associated subvolume propensity α_s , which is the sum of all propensities of all possible reaction events in that subvolume. Given the subvolume’s initial propensity $\alpha_{s,i}$, the time t_i until the first reaction event can be generated as

an exponential random variable via

$$t_i = -\frac{\log u}{\alpha_{s,i}}, \quad (\text{C.2})$$

where u is a uniform random number between 0 and 1. We sort the subvolumes in the indexed priority queue according to their reaction times, such that the subvolume with the lowest time is at the front of the queue (and can be immediately found, i.e., in $\mathcal{O}(1)$ time). When a reaction occurs in subvolume s , then the propensity $\alpha_{s,i}$ is updated (if the number of molecules of any type changed) and the time until the next subvolume event is obtained via (C.2). Sometimes, the propensity can change *without* a reaction occurring in the subvolume, i.e., if molecules are added to the subvolume by an active actor or via diffusion. If this occurs, then the reaction time can be updated *without* generating a new random number. Given the initial propensity $\alpha_{s,i}$, the updated propensity $\alpha_{s,f}$, the original reaction time t_i , and the current simulation time t_c , then the updated subvolume event time t_f is [77, p. 1881]

$$t_f = t_c + \frac{\alpha_{s,i}}{\alpha_{s,f}} (t_i - t_c), \quad (\text{C.3})$$

and this update is much faster than using (C.2). When the subvolume reaction time is updated (either via (C.2) or (C.3)), we also update the subvolume’s place in the indexed priority queue, which takes $\mathcal{O}(\log N_{\text{Sub}})$ time, where N_{Sub} is the total number of subvolumes. However, as noted in [77], updating the queue often takes much less than $\mathcal{O}(\log N_{\text{Sub}})$ time if a small number of subvolumes are much more likely to have an event than the others. At the end of the mesoscopic simulation step, we read the time associated with the subvolume at the front of the queue to update the mesoscopic timer.

Appendix C.4.2. Direct Method in NSM

The NSM uses the DM to determine which subvolume reaction takes place. This process is effectively a biased die roll where the probability of each event is proportional to the propensity of that event. If reaction c has propensity α_c , then it has probability α_c/α_s of being the next reaction in subvolume s . We perform the die roll by generating a uniform random number u and comparing it with the cumulative sum of α_c/α_s terms for subvolume s ; see [46, Eq. (27b)].

Using the DM takes $\mathcal{O}(N_{\text{Event}})$ time, where N_{Event} is the total number of possible events in that subvolume, i.e., all chemical reactions and diffusion of each type of molecule to every neighboring subvolume. Every distinguishable reaction is a distinct event. For example, consider a subvolume with two types of molecules. If the molecules can participate in one of three chemical reactions (either as a reactant or product), and both types of molecule can diffuse into any of the subvolume’s six neighbors, then there are a total of $3 + 2 \times 6 = 15$ possible events.

Appendix C.4.3. Executing Events

Once the NSM has determined the current event, it is executed by adjusting the corresponding molecule counts. This takes $\mathcal{O}(1)$ time for a diffusion event and $\mathcal{O}(|\Omega_{\text{Mol}}|)$ time for a chemical reaction event, where $|\Omega_{\text{Mol}}|$ is the number of molecule types. It also takes $\mathcal{O}(N_{\text{Event}})$ time to update the event propensities within the subvolume (or subvolumes in the case of diffusion). As mentioned previously, the individual event propensities are added to get the total subvolume propensity α_s and then we can recalculate the subvolume's next event time.

Given the time to execute the NSM and the time to execute the current event, one mesoscopic step takes $\mathcal{O}(\log N_{\text{Sub}} + N_{\text{Event}} + |\Omega_{\text{Mol}}|) = \mathcal{O}(\log N_{\text{Sub}} + N_{\text{Event}})$ time, since the number of events is a linear function of the number of types of molecules.

Appendix D. Review of Statistics

In this appendix, we review expressions for the probability distributions and mutual information used in AcCoRD's post-processing utilities.

Appendix D.1. Binomial Distribution

If we assume that all molecules behave independently and have the same probability of being observed at some time by a passive actor, then it is natural to describe the number of molecules observed by the actor at some time as a Binomial random variable; see [58, Ch. 5]. Suppose that X is a Binomial random variable and corresponds to the number of molecules observed. If n is the total number of molecules, and p is probability of a given molecule being observed, then the probability mass function (PMF) of X is [58, Eq. (5.1.2)]

$$\Pr\{X = k\} = \binom{n}{k} p^k (1-p)^{n-k}, \quad (\text{D.1})$$

where $k \in \{0, 1, \dots, n\}$. The PMF can be more convenient to calculate via [78, Eq. (26.5.25)]

$$\Pr\{X = k\} = I_p(i, n-i+1) - I_p(i+1, n-i), \quad (\text{D.2})$$

where [78, Eq. (6.6.2)]

$$I_x(a, b) = \frac{B_x(a, b)}{B(a, b)} \quad (\text{D.3})$$

is the regularized incomplete Beta function, [78, Eq. (6.6.1)]

$$B_x(a, b) = \int_0^x t^{a-1} (1-t)^{b-1} dt \quad (\text{D.4})$$

is the incomplete Beta function, and [78, Eq. (6.2.1)]

$$B(a, b) = B_x(a, b)|_{x=1}, \quad (\text{D.5})$$

is the Beta function. The cumulative distribution function (CDF) of X is [58, Eq. (5.1.4)]

$$\Pr\{X \leq i\} = \sum_{k=0}^i \binom{n}{k} p^k (1-p)^{n-k}, \quad (\text{D.6})$$

which can be more convenient to calculate via [78, Eq. (26.5.24)]

$$\Pr\{X \leq i\} = I_{1-p}(n-i, i+1). \quad (\text{D.7})$$

Appendix D.2. Poisson Approximation

One approximation of the Binomial distribution uses the Poisson distribution with mean np , which is valid when n is sufficiently large and p is sufficiently small; see [58, Ch. 5]. If X is a Poisson random variable with mean np , then its PMF is [58, Eq. (5.2.1)]

$$\Pr\{X = k\} = \frac{(np)^k \exp(-np)}{k!}, \quad (\text{D.8})$$

and its CDF is

$$\Pr\{X \leq i\} = \sum_{k=0}^i \frac{(np)^k \exp(-np)}{k!}. \quad (\text{D.9})$$

Using [78, Eqs. (26.4.19), (26.4.21)], it can be shown that the CDF can also be calculated as

$$\Pr\{X \leq i\} = \frac{\Gamma(\lfloor i+1 \rfloor, np)}{\Gamma(\lfloor i+1 \rfloor)}, \quad (\text{D.10})$$

where [78, Eq. (6.5.3)]

$$\Gamma(a, x) = \int_x^\infty t^{a-1} \exp(-t) dt \quad (\text{D.11})$$

is the incomplete Gamma function and [78, Eq. (6.1.1)]

$$\Gamma(a) = \Gamma(a, x)|_{x=0} \quad (\text{D.12})$$

is the Gamma function.

Appendix D.3. Gaussian Approximation

By the central limit theorem, a Binomial random variable can also be approximated as a Gaussian random variable with mean np and variance $np(1-p)$; see [58, Ch. 6]. However, since a Gaussian distribution has real support, we should include a continuity correction to approximate the discrete Binomial distribution. If Binomial random variable X is approximated by Gaussian random variable Y , then the PMF of X , $\Pr\{X = k\}$, should be approximated using $\Pr\{k-0.5 < Y < k+0.5\}$. By adding the continuity correction to [79, Eq. (2.1-93)], we approximate the CDF of Binomial random variable X as

$$\Pr\{X \leq k\} = \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{k+0.5-np}{\sqrt{2np(1-p)}} \right) \right], \quad (\text{D.13})$$

and we obtain the PMF $\Pr\{X = k\}$ by subtracting $\Pr\{X \leq k-1\}$ from $\Pr\{X \leq k\}$.

Appendix D.4. Mutual Information

The mutual information between two variables X and Y , $I(X;Y)$, measures how much the knowledge of one variable aids in the prediction of a second variable. It is defined as [80, Eq. (2.28)]

$$I(X;Y) = \sum_x \sum_y \Pr(X=x, Y=y) \times \log \frac{\Pr(X=x, Y=y)}{\Pr(X=x) \Pr(Y=y)}, \quad (\text{D.14})$$

where $\Pr(X=x, Y=y)$ is the joint probability distribution of X and Y , and $\Pr(X=x)$ and $\Pr(Y=y)$ are the marginal distributions of X and Y , respectively. Given realizations of the two variables X and Y , we can calculate these distributions empirically. If X and Y are independent, then knowing X does not provide any greater certainty in predicting the value of Y (or vice versa). Theoretically, independent variables have $I(X;Y) = 0$, though calculating mutual information from a finite number of realizations will generally result in a non-zero value; see [62].

Appendix E. Video Summary

There are eight videos in [23]. Video 1 shows a sample environment described in Section 3. The environments in the remaining videos are shown in Section 1 and complete descriptions can be found in Section 7. For clarity, each video generally shows fewer molecules than the corresponding simulation. Summaries of the video files are as follows:

1. **Video 1:** This video shows a simulation of the environment in Fig. 5 on page 8. Molecules are released by the blue surface. They can diffuse (in one direction) through the green membrane surface and be absorbed by the sphere. All other boundaries are reflective. The trajectory of one molecule is emphasized by showing it in red (until it is absorbed and turns blue).
2. **Video 2:** This video shows a simulation of System 1, whose environment is shown in Fig. 1 on page 3 and Fig. 8 on page 21. The system has adjacent microscopic and mesoscopic regions of equal size. Molecules are initialized uniformly throughout the environment and freely diffuse throughout. All boundaries are reflective. The trajectory of one molecule is emphasized by showing it in blue in the mesoscopic regime and in red in the microscopic regime.
3. **Video 3:** This video shows a simulation of a subset of System 2, whose environment is shown in Fig. 2 on page 3. The system is a rectangular rod that is mostly a mesoscopic region but the end with an absorbing surface is microscopic. Molecules are initialized uniformly throughout the environment and freely diffuse until they are absorbed. The molecules

are dark grey in the mesoscopic region, light grey in the microscopic region, and turn red when absorbed.

4. **Video 4:** This video shows a simulation of a small subset of System 3, whose full environment is shown in Fig. 3 on page 4. Molecules are initialized in the narrow space between two spheres. These molecules can irreversibly absorb to the outer sphere (and turn blue) and are reflected off of the inner sphere.
5. **Video 5:** This video shows a simulation of a small subset of System 3, whose full environment is shown in Fig. 3 on page 4. Molecules are initialized in the narrow space between two spheres. These molecules can probabilistically diffuse through the inner sphere in either direction (i.e., the inner sphere is a bidirectional membrane) and are reflected off of the outer sphere. The molecules turn yellow while they are inside the inner sphere.
6. **Video 6:** This video shows a simulation of System 4, whose environment is shown in Fig. 4 on page 4. A point transmitter releases molecules that are observed at a passive sphere. The released molecules are blue while they diffuse freely, yellow while they are bound to enzyme molecules, and red while they are inside the sphere. The unbound enzymes are white.
7. **Video 7:** This video shows a simulation of System 4, whose environment is shown in Fig. 4 on page 4. A point transmitter releases multiple pulses of molecules that are observed at a passive sphere. The molecules turn red while they are inside the sphere.
8. **Video 8:** This video shows a simulation of System 4, whose environment is shown in Fig. 4 on page 4. A point transmitter releases molecules that are observed at an absorbing sphere. The molecules turn red when they are absorbed.

References

References

- [1] S. Hiyama, Y. Moritani, T. Suda, R. Egashira, A. Enomoto, M. J. Moore, T. Nakano, Molecular communication, in: Proc. NSTI Nanotech, 2005, pp. 391–394.
- [2] L. C. M. Antunes, R. B. R. Ferreira, Intercellular communication in bacteria., Crit. Rev. Microbiol. 35 (2) (2009) 69–80. doi:10.1080/10408410902733946.
- [3] D. E. Sadava, D. M. Hillis, H. C. Heller, M. Berenbaum, Life: The Science of Biology, 10th Edition, Sinauer Associates, 2014.
- [4] B. Alberts, D. Bray, K. Hopkin, A. D. Johnson, J. Lewis, M. Raff, K. Roberts, P. Walter, Essential Cell Biology, 3rd Edition, Garland Science, 2009.
- [5] T. Nakano, A. W. Eckford, T. Haraguchi, Molecular Communication, Cambridge University Press, 2013.
- [6] N. Farsad, H. B. Yilmaz, A. Eckford, C.-B. Chae, W. Guo, A comprehensive survey of recent advancements in molecular communication, IEEE Commun. Surv. Tutorials 18 (3) (2016) 1887–1919. doi:10.1109/COMST.2016.2527741.
- [7] J. Crank, The Mathematics of Diffusion, 2nd Edition, Oxford University Press, 1979.

- [8] E. L. Cussler, *Diffusion: Mass Transfer in Fluid Systems*, Cambridge University Press, 1984.
- [9] H. C. Berg, *Random Walks in Biology*, Princeton University Press, 1993.
- [10] G. A. Truskey, F. Yuan, D. F. Katz, *Transport Phenomena in Biological Systems*, 2nd Edition, Pearson Prentice Hall, 2009.
- [11] R. Chang, *Physical Chemistry for the Biosciences*, University Science Books, 2005.
- [12] N. Farsad, W. Guo, A. W. Eckford, Tabletop molecular communication: Text messages through chemical signals, *PLoS One* 8 (12) (2013) e82935. doi:10.1371/journal.pone.0082935.
- [13] A. Noel, K. C. Cheung, R. Schober, Improving receiver performance of diffusive molecular communication with enzymes, *IEEE Trans. Nanobiosci.* 13 (1) (2014) 31–43. doi:10.1109/TNB.2013.2295546.
- [14] N. Farsad, N.-R. Kim, A. W. Eckford, C.-B. Chae, Channel and noise models for nonlinear molecular communication systems, *IEEE J. Sel. Areas Commun.* 32 (12) (2014) 2392–2401. doi:10.1109/JSAC.2014.2367662.
- [15] S. S. Andrews, D. Bray, Stochastic simulation of chemical reactions with spatial resolution and single molecule detail, *Phys. Biol.* 1 (3-4) (2004) 137–151. doi:10.1088/1478-3967/1/3/001.
- [16] G. Wei, P. Bogdan, R. Marculescu, Efficient modeling and simulation of bacteria-based nanonetworks with BNSim, *IEEE J. Sel. Areas Commun.* 31 (12) (2013) 868–878. doi:10.1109/JSAC.2013.SUP2.12130019.
- [17] L. Felicetti, M. Femminella, G. Reali, Simulation of molecular signaling in blood vessels: Software design and application to atherogenesis, *Nano Commun. Net.* 4 (3) (2013) 98–119. doi:10.1016/j.nancom.2013.06.002.
- [18] I. Llatser, D. Demiray, A. Cabellos-Aparicio, D. T. Altılar, E. Alarcón, N3Sim: Simulation framework for diffusion-based molecular communication nanonetworks, *Simul. Model. Pract. Theory* 42 (2014) 210–222. doi:10.1016/j.simpat.2013.11.004.
- [19] H. B. Yilmaz, C.-B. Chae, Simulation study of molecular communication systems with an absorbing receiver: Modulation and ISI mitigation techniques, *Simul. Model. Pract. Theory* 49 (2014) 136–150. doi:10.1016/j.simpat.2014.09.002.
- [20] Y. Jian, B. Krishnaswamy, C. M. Austin, A. O. Bicen, J. E. Perdomo, S. C. Patel, I. F. Akyildiz, C. R. Forest, R. Sivakumar, nanoNS3: Simulating bacterial molecular communication based nanonetworks in Network Simulator 3, in: *Proc. ACM NANOCOM*, 2016, pp. 1–7. doi:10.1145/2967446.2967464.
- [21] A. Noel, AcCoRD (Actor-based Communication via Reaction-Diffusion).
URL <https://github.com/adamjgnoel/AcCoRD/>
- [22] A. Noel, K. C. Cheung, R. Schober, Multi-scale stochastic simulation for diffusive molecular communication, in: *Proc. IEEE ICC*, 2015, pp. 1109–1115. doi:10.1109/ICC.2015.7248471.
- [23] A. Noel, Videos for AcCoRD journal paper (2016).
URL <https://www.youtube.com/watch?v=7QcN6eGrC4w&list=PLZ7uYXG-7XF8UyhFrIuQIiZig1XA89e3i>
- [24] S. S. Andrews, Accurate particle-based simulation of adsorption, desorption and partial transmission, *Phys. Biol.* 6 (4) (2009) 046015. doi:10.1088/1478-3975/6/4/046015.
- [25] A. Noel, D. Makrakis, A. Hafid, Channel impulse responses in diffusive molecular communication with spherical transmitters, in: *Proc. CSIT BSC* 2016, to appear, pp. 1–6.
- [26] A. Noel, K. C. Cheung, R. Schober, On the statistics of reaction-diffusion simulations for molecular communication, in: *Proc. ACM NANOCOM*, 2015, pp. 1–6. doi:10.1145/2800795.2800821.
- [27] S. Plimpton, Fast parallel algorithms for short-range molecular dynamics, *J. Comput. Phys.* 117 (1) (1995) 1–19. doi:10.1006/jcph.1995.1039.
- [28] S. Plimpton, LAMMPS molecular dynamics simulator.
URL <http://lammps.sandia.gov>
- [29] S. S. Andrews, Smoldyn: A spatial stochastic simulator for chemical reaction networks.
URL <http://www.smoldyn.org>
- [30] J. S. van Zon, P. R. ten Wolde, Green’s-function reaction dynamics: A particle-based approach for simulating biochemical networks in time and space, *J. Chem. Phys.* 123 (23) (2005) 234910. doi:10.1063/1.2137716.
- [31] R. Ramaswamy, I. F. Sbalzarini, Exact on-lattice stochastic reaction-diffusion simulations using partial-propensity methods., *J. Chem. Phys.* 135 (24) (2011) 244103. doi:10.1063/1.3666988.
- [32] B. Drawert, S. Engblom, A. Hellander, URDME: A modular framework for stochastic simulation of reaction-transport processes in complex geometries, *BMC Syst. Biol.* 6 (1) (2012) 76. doi:10.1186/1752-0509-6-76.
- [33] B. Bayati, P. Chatelain, P. Koumoutsakos, Adaptive mesh refinement for stochastic reaction-diffusion processes, *J. Comput. Phys.* 230 (1) (2011) 13–26. doi:10.1016/j.jcp.2010.08.035.
- [34] D. T. Gillespie, Approximate accelerated stochastic simulation of chemically reacting systems, *J Chem Phys* 115 (4) (2001) 1716 – 1733. doi:10.1063/1.1378322.
- [35] K. A. Iyengar, L. A. Harris, P. Clancy, Accurate implementation of leaping in space: The spatial partitioned-leaping algorithm, *J. Chem. Phys.* 132 (9) (2010) 094101. doi:10.1063/1.3310808.
- [36] COMSOL Inc., COMSOL Multiphysics.
URL <http://www.comsol.com>
- [37] ANSYS Inc., ANSYS.
URL <http://www.ansys.com>
- [38] M. Klann, A. Ganguly, H. Koepl, Hybrid spatial Gillespie and particle tracking simulation, *Bioinformatics* 28 (18) (2012) i549–i555. doi:10.1093/bioinformatics/bts384.
- [39] M. B. Flegg, S. J. Chapman, L. Zheng, R. Erban, Analysis of the two-regime method on square meshes, *SIAM J. Sci. Comput.* 36 (3) (2014) 561–588. doi:10.1137/130915844.
- [40] A. Hellander, S. Hellander, P. Lötstedt, Coupled mesoscopic and microscopic simulation of stochastic reaction-diffusion processes in mixed dimensions, *Multiscale Model. Simul.* 10 (2) (2012) 585–611. doi:10.1137/110832148.
- [41] M. Robinson, S. S. Andrews, R. Erban, Multiscale reaction-diffusion simulations with smoldyn, *Bioinformatics* 31 (14) (2015) 2406–2408. doi:10.1093/bioinformatics/btv149.
- [42] G. Wagner, R. Jones, J. Templeton, M. Parks, An atomistic-to-continuum coupling method for heat transfer in solids, *Comput. Methods Appl. Mech. Eng.* 197 (41) (2008) 3351–3365. doi:10.1016/j.cma.2008.02.004.
- [43] D. C. Resasco, F. Gao, F. Morgan, I. L. Novak, J. C. Schaff, B. M. Slepchenko, Virtual Cell: Computational tools for modeling in cell biology, *Wiley Interdiscip. Rev. Syst. Biol. Med.* 4 (2) (2012) 129–140. doi:10.1002/wsbm.165.
- [44] E. Gul, B. Atakan, O. B. Akan, NanoNS: A nanoscale network simulator framework for molecular communications, *Nano Commun. Net.* 1 (2) (2010) 138–156. doi:10.1016/j.nancom.2010.08.003.
- [45] A. Akkaya, G. Genc, T. Tugcu, HLA based architecture for molecular communication simulation, *Simul. Model. Pract. Theory* 42 (2014) 163–177. doi:10.1016/j.simpat.2013.12.012.
- [46] D. T. Gillespie, A general method for numerically simulating the stochastic time evolution of coupled chemical reactions, *J. Comput. Phys.* 22 (4) (1976) 403–434. doi:10.1016/0021-9991(76)90041-3.
- [47] D. T. Gillespie, Stochastic simulation of chemical kinetics., *Annu. Rev. Phys. Chem.* 58 (2007) 35–55. doi:10.1146/annurev.physchem.58.032806.104637.
- [48] I. Llatser, I. Pascual, N. Garralda, A. Cabellos-Aparicio, M. Pierobon, E. Alarcón, J. Solé Paretá, Exploring the physical channel of diffusion-based molecular communication by simulation, in: *Proc. IEEE GLOBECOM*, 2011, pp. 1–5. doi:10.1109/GLOCOM.2011.6134028.
- [49] I. Llatser, N3Sim: A simulation framework for diffusion-based molecular communication.
URL <http://www.n3cat.upc.edu/n3sim>
- [50] S. F. Bush, J. L. Paluh, G. Piro, V. Rao, R. V. Prasad, A. W. Eckford, Defining communication at the bottom, *IEEE Trans.*

- Mol. Biol. Multi-Scale Commun. 1 (1) (2015) 90–96. doi:10.1109/TBMC.2015.2465513.
- [51] M. Matsumoto, T. Nishimura, Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator, *ACM Trans. Model. Comput. Simul.* 8 (1) (1998) 3–30. doi:10.1145/272991.272995.
- [52] M. E. O’Neill, PCG: A family of simple fast space-efficient statistically good algorithms for random number generation, *ACM Trans. Math. Softw.* (submitted) 1–46.
- [53] M. S. Kuran, H. B. Yilmaz, T. Tugcu, I. F. Akyildiz, Modulation techniques for communication via diffusion in nanonetworks, in: *IEEE Int. Conf. Commun.*, 2011, pp. 1–5. doi:10.1109/icc.2011.5962989.
- [54] R. H. Petrucci, F. G. Herring, J. D. Madura, C. Bissonnette, *General Chemistry: Principles and Modern Applications*, 11th Edition, Pearson Prentice Hall, 2016.
- [55] D. Crockford, JSON: JavaScript Object Notation. URL <http://www.json.org>
- [56] D. Bernstein, Simulating mesoscopic reaction-diffusion systems using the gillespie algorithm, *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.* 71 (4) (2005) 1–13. doi:10.1103/PhysRevE.71.041103.
- [57] E. W. Ng, M. Geller, A table of integrals of the error functions, *J. Res. Natl. Bur. Stand. Sect. B Math. Sci.* 73B (1) (1969) 1–20. doi:10.6028/jres.073B.001.
- [58] S. M. Ross, *Introduction to Probability and Statistics for Engineers and Scientists*, 4th Edition, Academic Press, 2009.
- [59] S. Hellander, L. Petzold, Reaction rates for a generalized reaction-diffusion master equation, *Phys. Rev. E* 93 (1) (2016) 13307. doi:10.1103/PhysRevE.93.013307.
- [60] R. Huang, I. Chavez, K. M. Taute, B. Lukić, S. Jeney, M. G. Raizen, E.-L. Florin, Direct observation of the full transition from ballistic to diffusive brownian motion in a liquid, *Nat. Phys.* 7 (7) (2011) 576–580. doi:10.1038/nphys1953.
- [61] A. Noel, K. C. Cheung, R. Schober, Optimal receiver design for diffusive molecular communication with flow and additive noise, *IEEE Trans. Nanobiosci.* 13 (3) (2014) 350–362. doi:10.1109/TNB.2014.2337239.
- [62] B. Goebel, Z. Dawy, J. Hagenauer, J. Mueller, An approximation to the distribution of finite sample size mutual information estimates, in: *Proc. IEEE ICC*, 2005, pp. 1102–1106. doi:10.1109/ICC.2005.1494518.
- [63] Y. Deng, A. Noel, W. Guo, A. Nallanathan, M. ElKashlan, Stochastic geometry model for large-scale molecular communication systems, in: *Proc. IEEE GLOBECOM 2016*, to appear, pp. 1–7.
- [64] A. Noel, Y. Deng, D. Makrakis, A. Hafid, Active versus passive: Receiver model transforms for diffusive molecular communication, in: *Proc. IEEE GLOBECOM 2016*, to appear, pp. 1–6.
- [65] A. C. Heren, H. B. Yilmaz, C.-B. Chae, T. Tugcu, Effect of degradation in molecular communication: Impairment or enhancement?, *IEEE Trans. Mol. Biol. Multi-Scale Commun.* 1 (2) (2015) 217–229. doi:10.1109/TBMC.2015.2502859.
- [66] L.-S. Meng, P.-C. Yeh, K.-C. Chen, I. F. Akyildiz, On receiver design for diffusion-based molecular communication, *IEEE Trans. Signal Process.* 62 (22) (2014) 6032–6044. doi:10.1109/TSP.2014.2359644.
- [67] M. U. Mahfuz, D. Makrakis, H. T. Mouftah, A comprehensive analysis of strength-based optimum signal detection in concentration-encoded molecular communication with spike transmission, *IEEE Trans. Nanobiosci.* 14 (1) (2015) 67–83. doi:10.1109/TNB.2014.2368593.
- [68] H. ShahMohammadian, G. G. Messier, S. Magierowski, Optimum receiver for molecule shift keying modulation in diffusion-based molecular communication channels, *Nano Commun. Net.* 3 (3) (2012) 183–195. doi:10.1016/j.nancom.2012.09.006.
- [69] A. Noel, K. C. Cheung, R. Schober, Using dimensional analysis to assess scalability and accuracy in molecular communication, in: *Proc. IEEE ICC MoNaCom*, 2013, pp. 818–823. doi:10.1109/ICCW.2013.6649346.
- [70] K. Schulten, I. Kosztin, *Lectures in Theoretical Biophysics*, University of Illinois, 2000.
- [71] P. Nelson, *Biological Physics: Energy, Information, Life*, Updated 1st Edition, W. H. Freeman and Company, 2008.
- [72] N. Farsad, A. Goldsmith, A molecular communication system using acids, bases and hydrogen ions, in: *Proc. IEEE SPAWC*, 2016, pp. 1–6. doi:10.1109/SPAWC.2016.7536834.
- [73] A. Ahmadzadeh, A. Noel, R. Schober, Analysis and design of multi-hop diffusion-based molecular communication networks, *IEEE Trans. Mol. Biol. Multi-Scale Commun.* 1 (2) (2015) 144–157. doi:10.1109/TBMC.2015.2501741.
- [74] Z. Jedrzejewski-Szmek, K. T. Blackwell, Asynchronous tau-leaping, *J. Chem. Phys.* 144 (12) (2016) 125104. doi:10.1063/1.4944575.
- [75] R. Erban, S. J. Chapman, Reactive boundary conditions for stochastic simulations of reaction-diffusion processes., *Phys. Biol.* 4 (1) (2007) 16–28. doi:10.1088/1478-3975/4/1/003.
- [76] J. Elf, M. Ehrenberg, Spontaneous separation of bi-stable biochemical systems into spatial domains of opposite phases, *Syst. Biol. (Stevenage)*. 1 (2) (2004) 230–236. doi:10.1049/sb:20045021.
- [77] M. A. Gibson, J. Bruck, Efficient exact stochastic simulation of chemical systems with many species and many channels, *J. Phys. Chem. A* 104 (9) (2000) 1876–1889. doi:10.1021/jp993732q.
- [78] M. Abramowitz, I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, 1st Edition, United States Department of Commerce, National Bureau of Standards, 1964.
- [79] J. G. Proakis, *Digital Communications*, 4th Edition, McGraw-Hill, 2000.
- [80] T. M. Cover, J. A. Thomas, *Elements of Information Theory*, 2nd Edition, Wiley-Interscience, 2006.



Adam Noel received the B.Eng. degree in Electrical Engineering in 2009 from Memorial University in St. Johns, Canada and the M.A.Sc. and Ph.D. degrees in Electrical and Computer Engineering from the University of British Columbia in Vancouver, Canada, in 2011 and 2015, respectively. In 2013, he was a Visiting Scientist at the Institute for Digital Communication at Friedrich-Alexander-University in Erlangen, Germany. He is currently a Postdoctoral Fellow at the University of Ottawa, Canada. Dr. Noel’s current research interests include channel modeling, system design, and simulation methods for molecular communication networks.



Karen C. Cheung received the B.S. and Ph.D. degrees in bioengineering from the University of California, Berkeley, in 1998 and 2002, respectively. From

2002 to 2005, she was a postdoctoral researcher at the Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland. She is now at the University of British Columbia, Vancouver, BC, Canada. Her research interests include lab-on-a-chip systems for cell culture and characterization, inkjet printing for tissue engineering, and implantable neural interfaces.

agement of next generation networks, QoS management, distributed systems, and communication protocols.



Robert Schober received the Diplom and the Ph.D. degrees in electrical engineering from Friedrich-Alexander-University (FAU), Erlangen, Germany, in 1997 and 2000, respectively. Since May 2002 he has been with the University of British Columbia, Vancouver, Canada, where he is now a Full Professor. Since January 2012 he is an Alexander von Humboldt Professor and the Chair for Digital Communication at FAU. His research interests fall into the broad areas of Communication Theory, Wireless Communications, and Statistical Signal Processing. From 2012 to 2015, he was the Editor-in-Chief of the IEEE Transactions on Communications.

Professor Dimitrios Makrakis is currently with the School of Electrical Engineering and Computer Science (EECS) of the University of Ottawa, and is the founding Director of the Broadband and Wireless Research Laboratory (BroadWIRLab). Prior to joining the University of Ottawa, Dr. Makrakis was a faculty member at the Department of Electrical Engineering of Western University (prior the University of Western Ontario) and Adjunct Professor at the Computer Science Department of Western U. His research evolves around the subject of information transfer, computing and cyber security and is both theoretical and experimental.



Abdelhakim Hafid is Full Professor at the University of Montreal, where he founded the Network Research Lab (NRL) in 2005. He is also a research fellow at CIRRELT (Interuniversity Research Center on Enterprise Networks, Logistics and Transportation). He supervised to graduation more than 30 graduate students. He published over 200 journal and conference papers, and he also holds 3 US patents. Dr. A. Hafid has extensive academic and industrial research experience in the area of the man-