

eGFRD in all dimensions

Thomas R. Sokolowski^{*†1}, Joris Paijmans¹, Laurens Bossen¹, Martijn Wehrens¹,
 Thomas Miedema¹, Nils B. Becker^{‡1}, Kazunari Kaizu², Koichi Takahashi²,
 Marileen Dogterom^{§1}, and Pieter Rein ten Wolde¹

¹FOM Institute AMOLF, Science Park 104, 1098 XG Amsterdam, The Netherlands

²RIKEN Quantitative Biology Center (QBIC), RIKEN, 6-2-3 Furuedai, Suita,
 Osaka 565-0874, Japan

July 31, 2018

Abstract

Biochemical reactions typically occur at low copy numbers, but at once in crowded and diverse environments. Space and stochasticity therefore play an essential role in biochemical networks. Spatial-stochastic simulations have become a prominent tool for understanding how stochasticity at the microscopic level influences the macroscopic behavior of such systems. However, while particle-based models guarantee the level of detail necessary to accurately describe the microscopic dynamics at very low copy numbers, the algorithms used to simulate them oftentimes imply trade-offs between computational efficiency and biochemical accuracy. eGFRD (enhanced Green's Function Reaction Dynamics) is an exact algorithm that evades such trade-offs by partitioning the N -particle system into $M \leq N$ analytically tractable one- and two-particle systems; the analytical solutions (Green's functions) then are used to implement an event-driven particle-based scheme that allows particles to make large jumps in time and space while retaining access to their state variables at any moment. Here we present "eGFRD2", a new eGFRD version that implements the principle of eGFRD in all dimensions, thus enabling efficient simulation of biochemical reaction-diffusion processes in the 3D cytoplasm, on 2D planes representing membranes, and on 1D elongated cylinders representative of, e.g., cytoskeletal tracks or DNA; in 1D, it also incorporates convective motion used to model active transport. We find that, for low particle densities, eGFRD2 is up to 3 orders of magnitude faster than optimized Brownian Dynamics. We exemplify the capabilities of eGFRD2 by simulating an idealized model of Pom1 gradient formation, which involves 3D diffusion, active transport on microtubules, and autophosphorylation on the membrane, confirming recent results on this system and demonstrating that it can efficiently operate under genuinely stochastic conditions.

^{*}Corresponding author; e-mail: tsokolowski@ist.ac.at

[†]Present address: Institute of Science and Technology (IST) Austria, Am Campus 1, 3400 Klosterneuburg, Austria

[‡]Present address: Bioquant Center, University of Heidelberg, 69120 Heidelberg, Germany

[§]Present address: Department of Bionanoscience, Kavli Institute of Nanoscience, Faculty of Applied Sciences, Delft University of Technology, 2628 CJ Delft, The Netherlands

1 Introduction

Biochemical reactions constitute the basis of all vital functions in biological cells, ranging from metabolism and gene regulation to environment sensing and intra- and intercellular transport. While even the simplest biological cells contain a myriad of different biochemical species, their individual copy numbers oftentimes only reach numbers as low as thousands, or even dozens [1, 2, 3, 4, 5]; this means that specific biochemical reaction pathways usually operate in the extreme low-concentration regime, while at the same time the cytoplasm is a highly crowded and inhomogeneous environment [6, 7, 8, 9, 10, 11]. These circumstances strongly augment the importance of spatial effects and the inherent stochasticity of biochemical reactions and at once hinder their direct experimental observation [12, 13]. For example, spatial inhomogeneities can have a strong influence on the behavior of spatially distributed enzymes [14, 15], even provoking the emergence or destruction of ultrasensitivity [16, 17, 18, 19], and on density-dependent clustering [20, 21, 22]; macromolecular crowding can shift chemical equilibria [23, 24, 25] (see [26] for a review), and fast reactant rebinding can significantly enhance the noise in transcription factor and ligand binding [27, 28, 29]. Facilitated diffusion on one-dimensional submanifolds, such as the DNA or cytoskeletal macropolymers, is capable of enhancing the search for target sites [30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41]. Perhaps most strikingly, spatio-temporal fluctuations at the molecular scale can drastically change the macroscopic behavior on the cellular scale [18, 42, 43].

Spatial-stochastic simulations therefore have become an important tool for understanding biochemical mechanisms. They can roughly be separated into two classes: lattice- or mesh-based simulation schemes and particle-based schemes [44, 45]. Mesh-based schemes, such as *MesoRD* [46, 42, 47], *URDME* [48] and associated techniques [49, 50, 51] (which recently lead to the development of *StochSS* [52]), *VCell* [53] and *GMP* [54, 55], elaborate on the idea of the event-driven (and thus highly efficient) Stochastic Simulation Algorithm by Gillespie [56, 57], by essentially implementing it on a spatial mesh; therefore, as a caveat, they have to assume well-mixedness at least locally, which—in general—is an inaccurate representation of the real conditions in biological cells. Particle-based schemes such as *Smoldyn* [58, 59], *MCell* [60, 61, 62, 63, 64], *ChemCell* [65], *GridCell* [66], *Spatioocyte* [67, 68], and *ReaDDy* [69] are traditionally based on the principle of Brownian Dynamics (BD); here particle diffusion is approximated by a random walk in continuous space with very small propagation steps ($\Delta t \lesssim 10^{-6}s$), required to render them sufficiently accurate. At low concentrations these schemes become inefficient, because most CPU time is spent on generating (uninteresting) random movements; moreover, since their capability to sample chemical equilibria faithfully depends on how well particle overlaps are resolved, their computational efficiency can be only improved at the cost of sacrificing accuracy.

The desire to overcome this antagonism between efficiency and accuracy lead to the development of eGFRD (“enhanced Green’s Function Reaction Dynamics”) [70, 71, 27, 18], which is both particle-based and event-driven, and does not rely on arbitrary definitions of particle contact to sample reactions. The key idea of eGFRD is to partition the space filled by the particle cloud into geometrically simple subvolumes (“domains”) which contain at most two particles; after breaking down the multi-particle problem into a series of one- and two-particle problems, the full time-dependent analytical solution of the reaction-diffusion problem can be calculated for each of the domains, and used to sample exact event times and updated particle positions. This way, large jumps in time and space can be made by each individual particle, rendering eGFRD orders of magnitude more efficient than conventional BD schemes up to μM concentrations [18].

However, until now eGFRD had been limited to simulations of diffusion and particle interactions in three-dimensional space. Yet it is well known that many biochemical reactions occur on finite 1D and 2D submanifolds of the cell, such as the cell membrane, membranes of intracellular vesicles, and long macropolymers like the DNA or microtubules [72, 73, 74, 35, 9, 41, 75]. In this work we present “eGFRD2”, an extended version of eGFRD that allows for simulations in all dimensions, implementing diffusion and particle reactions in 1D and 2D, binding of bulk particles to lower dimensional structures, and transitions of particles between different structures; in 1D, it also features combined diffusive-convective motion with reactions, allowing for simulation of active transport on cytoskeletal tracks. To accomplish this, we derived and numerically implemented the Green’s functions by solving the one- and two-particle reaction-diffusion problems in 1D and 2D, and integrated them together with the known 3D functions and a BD fallback system into a new user-friendly simulation environment. In order to exemplify the possibilities of the new eGFRD we carried out simulations of Pom1 gradient formation, which is driven by autophosphorylation on the membrane and active transport, and were able to confirm recent results on this system, while demonstrating that it can operate efficiently under fundamentally stochastic conditions, owing to low copy numbers.

This paper is organized as follows: In the first part (“Methods”) we first recapitulate the working principle of eGFRD, followed by a description of the new extensions to lower dimensions, and a brief presentation of the performance of the new scheme. In the second part (“Results”) we introduce the studied example system and present our simulation results. We end by discussing the results and an outlook on further development.

2 Methods

2.1 The eGFRD working principle

eGFRD is an exact algorithm designed to simulate the idealized reaction-diffusion model shown in Figure 1, which is widespread in the field of particle-based stochastic simulation. In this “particle-based model”, the particles have an idealized, spherical shape with a species-specific radius R , move by normal free diffusion characterized by a (species-specific) diffusion constant D , and can interact upon contact with a predefined rate constant k ; eGFRD thus assumes that beyond the contact distance the interaction potential is zero. In addition, the particles can undergo dissociation, species change or annihilation reactions with predefined rates. Stochastic simulations of the particle-based model in Fig. 1 can be straightforwardly carried out using Brownian Dynamics, but at low particle density—commonly encountered in biochemical systems—this becomes very inefficient, because the vast majority of computation steps is spent on sampling the diffusive random walks of the particles; it is therefore desirable to skip the particle hops and jump directly between the truly interesting events, i.e., particle encounters and reactions, employing the known statistics of diffusion [76, 77, 78, 79, 80]. However—even with the simplifications introduced above—it is generally hard to find an analytical prediction for future particle species and positions in an N -particle reaction-diffusion system; nonetheless, as described further below in more detail, exact analytical solutions (Green’s functions) can be obtained for the case $N \leq 2$. eGFRD capitalizes on this fact by dividing the simulation volume into subvolumes, called protective domains, that contain either one (“*Single*” domains) or two (“*Pair*” domains) particles, in order to isolate the content of each domain from the influence of surrounding particles (and vice versa). This way the N -particle problem is reduced to $M \leq N$ independent one- or two-particle problems. Figure 1 illustrates this principle. When—after a domain-specific time $\tau_{\mathcal{D}}$ that can be sampled from the Green’s functions—one of the particles hits a domain boundary or experiences a reaction that changes its biochemical properties, the state of the involved particle(s) is updated, the old domain removed, and one or more new domains initialized. The use of protective domains is the key innovation of eGFRD compared to the original GFRD [70, 71, 27], which was based on the same motivation, but had to operate with a maximal cut-off time for particle updates in order to render particle interactions not captured by the used unbounded Green’s functions sufficiently improbable. Since in eGFRD by construction all position updates remain confined to the respective domain, any interference with the situation outside is not just improbable, but completely impossible. eGFRD therefore is an exact algorithm.

We will now describe how Green’s functions can be used to generate next-event times and corresponding new particle states within the protective domains in more detail. Let us first focus on the *Single* domain and denote by $p_1(\mathbf{r}, t|\mathbf{r}_0)$ the probability density function (PDF) for the diffusing particle being at position \mathbf{r} at time t , given that it started at position \mathbf{r}_0 . Then $p_1(\mathbf{r}, t|\mathbf{r}_0)$ is the Green’s function of the boundary value problem

$$\partial_t p_1(\mathbf{r}, t|\mathbf{r}_0) = D\nabla_{\mathbf{r}}^2 p_1(\mathbf{r}, t|\mathbf{r}_0) + \delta(\mathbf{r} - \mathbf{r}_0)\delta(t - t_0) \quad (1)$$

$$p_1(\mathbf{r}, t|\mathbf{r}_0) = 0 \quad \text{for} \quad \mathbf{r} \in \partial\mathcal{D}_1 \quad (2)$$

where the last equation imposes absorbing boundary conditions on the outer shell ($\partial\mathcal{D}_1$) of the domain. Note that here we do not specify the Laplace operator $\nabla_{\mathbf{r}}^2$ in detail yet; its precise form depends on the dimensionality of the underlying diffusion process.

Similarly, the exact solution for the PDF $p_2(\mathbf{r}_A, \mathbf{r}_B, t|\mathbf{r}_{A0}, \mathbf{r}_{B0})$, describing the positions \mathbf{r}_A and \mathbf{r}_B of two particles A and B inside a *Pair* domain after time t given that

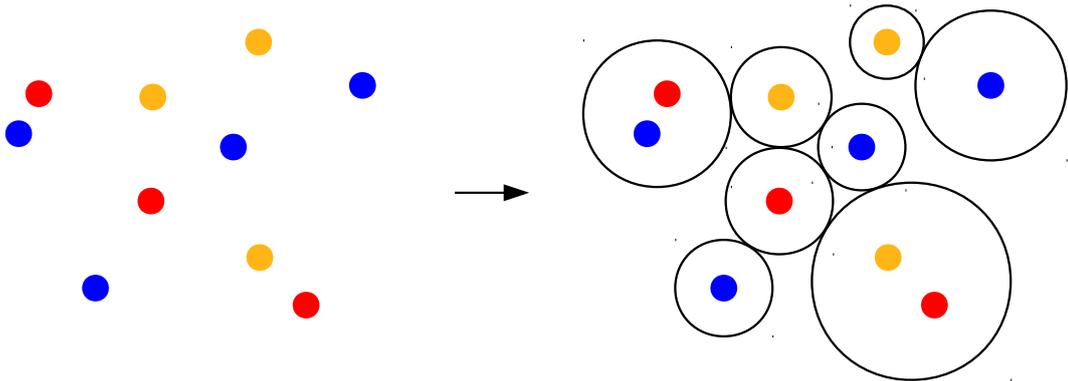


Figure 1: Protective domains separate the N -particle problem into one- and two-particle problems. The drawing illustrates how eGFRD constructs protective domains that contain at most two particles in order to isolate these from the influence of other particles, starting from a random particle configuration. Subsequently, analytical solutions are calculated for each domain individually and used to propagate the domains in an event-driven, asynchronous fashion. We show here a 2D projection for the standard scenario in which particles diffuse and react in unbounded 3D space. In this case protective domains are spherical. Different colors mark different chemical species.

they started at positions \mathbf{r}_{A0} and \mathbf{r}_{B0} , can be obtained by solving the SMOLUCHOWSKI equation [81, 71]

$$\partial_t p_2(\mathbf{r}_A, \mathbf{r}_B, t | \mathbf{r}_{A0}, \mathbf{r}_{B0}) = [D_A \nabla_{\mathbf{r}_A}^2 + D_B \nabla_{\mathbf{r}_B}^2] p_2(\mathbf{r}_A, \mathbf{r}_B, t | \mathbf{r}_{A0}, \mathbf{r}_{B0}) \quad (3)$$

after separating it into two individual diffusion equations for the interparticle vector \mathbf{r} and a “center of motion” coordinate \mathbf{R} via a product ansatz $p_2(\mathbf{r}_A, \mathbf{r}_B, t) \equiv p_r(\mathbf{r}, t) p_R(\mathbf{R}, t)$ (as described in detail in sec. S3 of the Supplementary Information), and imposing the following boundary conditions:

$$q_\sigma(t) \equiv - \int_{|\mathbf{r}|=\sigma} D \nabla_{\mathbf{r}} p_r(\mathbf{r}, t | \mathbf{r}_0) d\mathbf{r} = k p_r(|\mathbf{r}| = \sigma, t) \quad (4)$$

$$p_r(\mathbf{r}, t) = 0 \quad \text{for} \quad \mathbf{r} \in \partial \mathcal{D}_r \quad (5)$$

$$p_R(\mathbf{R}, t) = 0 \quad \text{for} \quad \mathbf{R} \in \partial \mathcal{D}_R \quad (6)$$

Here, Eq. (4), representing the particle reaction at intrinsic association rate k , imposes a radiating (flux) boundary condition to the interparticle vector \mathbf{r} at the particle contact radius $\sigma = R_A + R_B$, while Eq. (5) imposes an absorbing boundary condition at the outer radius of the “interparticle domain” $\partial \mathcal{D}_r$, and Eq. (6) does the same to the center-of-motion vector \mathbf{R} ; the subdomains $\partial \mathcal{D}_r$ and $\partial \mathcal{D}_R$ have to be chosen such that they fit inside the shell $\partial \mathcal{D}_2$ of the original *Pair* domain constructed around the two particles (this rule is exemplified in sec. S1.2.2 / Fig. S1A of the Supplementary Information). Importantly, since the form of the Laplacian and the precise form of the integral in the radiating condition Eq. (4) vary with the dimensionality of the problem, the Green’s functions are different in 1D, 2D and 3D and have to be calculated for each dimension separately.

Quantities that derive from the Green’s function, the survival probability and the boundary fluxes, can be used to generate tentative next-event times for each domain individually. Most importantly, since $p(\mathbf{r}, t | \mathbf{r}_0)$ completely describes the transient dynamics within the domain, it enables exact sampling of new particle positions at *any*

time t after domain construction, and at the next-event time $\tau_{\mathcal{D}}$ in particular, rendering incremental sampling of particle trajectories unnecessary. The procedure of sampling next-event times and new positions from the Green’s functions is described in detail in sec. S1.2 of the Supplementary Information. After each domain update the domain is removed, the new configuration of particles is reanalysed and new domains are constructed around the displaced particles. Then the newly calculated next-event times are inserted into the ordered scheduler in the right place, and the domain with the foremost next-event time is updated next. To enhance the formation of two-particle domains, recently updated particles can force a “premature” update of domains in their proximity, called “bursting”, by which the domain is propagated towards a time prior to its originally scheduled update. If bursting causes particles to move close enough, creation of a two-particle domain will be attempted. If there is not enough space to construct any eGFRD domain due to nearby obstacles or particle crowding, the particles are propagated by a Brownian Dynamics fallback simulator, as explained in sec. 2.3 and supplementary sec. S2. A compact overview of the basic eGFRD algorithm in pseudo-code is given by Algorithm 1 in the Supplementary Information (p. 2), while a detailed account of the bursting and domain construction rules is found in supplementary sec. S6.

2.2 Extension to lower dimensions

In order to port the eGFRD principle to lower dimensions we introduced static reactive surfaces capturing the essential geometric features of subcellular structures: finite planes, which can be used to model membranes, and (thin) finite cylinders, representative, e.g., of elongated DNA or cytoskeletal tracks (microtubules, actin filaments, etc.). Based on this we defined a new set of protective domains for interactions of particles *with* the new structures (“*Interaction*” domains) and new *Single* and *Pair* domains for diffusion and interparticle reactions *on* the structures, and calculated the Green’s functions for the associated reaction-diffusion problems within the respective geometry. Figures 2 and 3 contain an overview of the most important new domains in 2D and 1D, respectively; most of them are cylindrical, reflecting the natural coordinate separation for the respective binding or transport process.

Below we motivate and explain the principal new domain types in more detail, and briefly sketch the derivation of the associated new Green’s functions; for the complete mathematical derivations, sampling and domain making rules we will refer the reader to the Supplementary Information. Several domain types devised for special applications are described in sec. 2.2.6.

2.2.1 Binding to planes

Fig. 2A schematically shows the “*Planar Surface Interaction*” domain used for interactions of a bulk particle (undergoing 3D diffusion) with a reactive plane. While not strictly necessary, we chose a cylindrical geometry for the domain, because it facilitates its scaling with respect to the other cylindrical domains for plane-bound particles that we will introduce further below. The height of the domain over the plane $h = \delta + \beta R_0$ is composed of the particle-plane distance δ plus βR_0 , where R_0 is the particle radius and $\beta > 1$ a safety factor (the “single-shell factor”, defined in section S6 of the Supplementary Information); the domain radius is determined by the available space in the vicinity of the domain. Association to the plane is modeled via a radiating boundary condition at particle-plane contact, whereby the particle is defined to be at contact when its center touches the plane. For that reason, the domain slightly extends behind the reactive plane by a length $h' = \beta R_0$, to prevent the bound particle

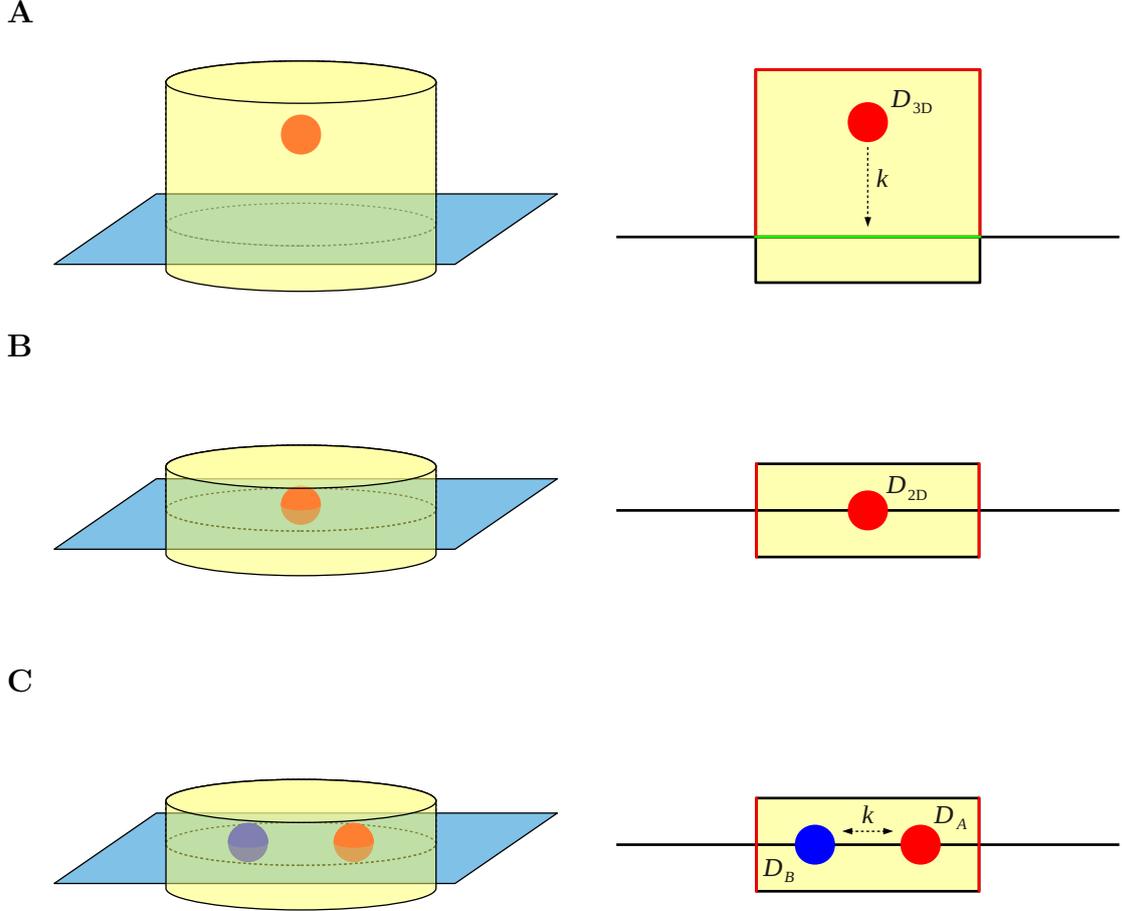


Figure 2: New protective domain types for interaction with and reaction-diffusion on planar surfaces. (A) *Planar Surface Interaction* domain; (B) *Planar Surface Single* domain; (C) *Planar Surface Pair* domain. Right panels show sections of 3D objects. Absorbing boundaries are highlighted by red, radiating boundaries by green.

from overlapping with particles on the opposite side of the plane. By default we allow for association from both sides of the plane.

The particle can exit the domain by either binding to the plane or by hitting one of the absorbing domain boundaries, i.e. the cylinder tube or the (more distant) cylinder cap. Let $p(r, \varphi, z, t|r_0, \varphi_0, z_0)$ be the probability density function for this problem, written in cylindrical coordinates $\mathbf{r} = (r, \varphi, z)$. We can separate diffusion along the cylinder axis (z) from diffusion in the polar plane (r, φ) via the ansatz

$$p(r, \varphi, z, t|r_0, \varphi_0, z_0) = p_r^A(r, \varphi, t|r_0, \varphi_0) p_z^{\text{RA}}(z, t|z_0) \quad (7)$$

which yields a one-dimensional diffusion equation for p_z^{RA} and BESSEL's equation for p_r^A . The 1D-problem for p_z^{RA} has to be solved with a radiating boundary at $z = 0$ and an absorbing boundary at $z = h$; this is a special case of the 1D Green's function that we portray below in sec. 2.2.4 and in supplementary sec. S4.1.2 (with $v = 0$, $\sigma = 0$ and $a = h$). The equation for p_r^A has perfect radial symmetry by construction and describes 2D diffusion in polar coordinates for a particle starting at $r = 0$ and

a circular absorbing boundary at $r = R$. The solution to this problem is well-known and presented in section S4.3 of the Supplementary Information.

From the two Green's functions p_r^A and p_z^{RA} we sample next-event times τ_r and τ_z in the usual way and take their minimum as the next-event time $\tau_\nu = \min(\tau_r, \tau_z)$ of the interaction domain. In the case $\tau_\nu = \tau_z$, i.e. when the particle exits the cylindrical domain through one of its caps, we compare the fluxes through the opposite boundaries to determine whether the particle left through the absorbing ("IV *Escape*" event, particle remains in the cytoplasm) or through the radiating boundary ("IV *Interaction*", particle associates with the plane). For $\tau_\nu = \tau_r$ we know with certainty that a (radial) escape through the cylinder tube occurred. In both cases the respective other coordinate is sampled from the corresponding Green's function normalized by the respective survival probability, following the principle explained in sec. S1.2.2 of the Supplementary Information.

2.2.2 2D-diffusion and reactions on the plane

Diffusion and interaction of particles bound to the plane is simulated using the *Planar Surface Single* and *Planar Surface Pair* domains, respectively, shown in Fig. 2B and Fig. 2C. In analogy to the spherical *Single* and *Pair* domains in 3D, the 2D domains are cylindrical; the height of their cylindrical shells is determined by the particle diameter (times safety factors), while the radius again is dependent on the available space in its surroundings.

In the *Planar Surface Single* the particle starts out from the center of the domain, and the only exit channel is the absorbing boundary at its outer radius. The Green's function for this problem is precisely the one that describes the polar movement in the *Planar Surface Interaction* domain, p_r^A (see supplementary sec. S4.3).

In the *Planar Surface Pair* we perform the coordinate transform initially described in sec. 2.1, partitioning the available space among the center-of-motion coordinate \mathbf{R} and the interparticle vector \mathbf{r} ; this way the reaction-diffusion process is once again separated into two independent diffusion processes, while the particle interaction can be completely characterized by a radiating boundary condition to the \mathbf{r} coordinate. Next-event times and new positions for the polar diffusion in the \mathbf{R} coordinate then can be sampled from the Green's function $p_r^A(R, \Phi, t|R_0, \Phi_0)$. For the interparticle coordinate \mathbf{r} we use the Green's function $p_r^{RA}(r, \varphi, t|r_0, \varphi_0)$, which solves the following boundary value problem:

$$\begin{aligned} \partial_t p_r^{RA}(r, \varphi, t|r_0, \varphi_0) &= D_r \nabla_{\mathbf{r}}^2 p_r^{RA}(r, \varphi, t|r_0, \varphi_0) \\ &= D_r \left[\partial_r^2 + \frac{1}{r} \partial_r + \frac{1}{r^2} \partial_\varphi^2 \right] p_r^{RA}(r, \varphi, t|r_0, \varphi_0) \end{aligned} \quad (8)$$

$$2\pi\sigma D_r \partial_r p_r^{RA}(r, \varphi, t|r_0, \varphi_0)|_{r=\sigma} = k p_r^{RA}(|\mathbf{r}| = \sigma|r_0, \varphi_0) \quad (9)$$

$$p_r^{RA}(r, \varphi, t|r_0, \varphi_0)|_{r=a} = 0 \quad (10)$$

$$p_r^{RA}(r, \varphi, t = 0|r_0, \varphi_0) = \frac{1}{r} \delta(r - r_0) \delta(\varphi - \varphi_0) \quad (11)$$

Herein, Eq. (8) is the diffusion equation in polar coordinates, Eq. (9) the radiating boundary condition modeling reactions at a contact radius $\sigma = R_A + R_B$ with intrinsic reaction rate k , Eq. (10) an absorbing boundary condition at the outer radius a of the interparticle subdomain, and Eq. (11) the initial condition of the interparticle vector properly transformed into polar coordinates. We solve this problem explicitly in sec. S4.2 of the Supplementary Information.

To determine the next event for the *Pair* domain, we first sample next-event times τ_r and τ_R for the interparticle and center-of-motion coordinates, respectively; the smaller of the two is taken to be the next-event time for the whole domain. If $\tau_R < \tau_r$, the corresponding event is an exit of the \mathbf{R} coordinate from its subdomain, such that its new length $|\mathbf{R}|$ is fixed (equal to the outer radius of the subdomain), but we still have to sample the corresponding new angle Φ , and moreover a new length and a new angle for the interparticle vector \mathbf{r} ; the latter is achieved by plugging the time τ_R into $p_r^{\text{RA}}(r, \varphi, t|r_0, \varphi_0)$. If, conversely, $\tau_r < \tau_R$, we can sample a new \mathbf{R} vector in a similar way from $p_r^{\text{A}}(R, \Phi, \tau_r|R_0, \Phi_0)$. However, now concerning the dynamics of \mathbf{r} , two events are possible: either the \mathbf{r} coordinate hit the inner boundary and reacted at interparticle contact (“*IV Reaction*”), or it left the \mathbf{r} -subdomain at its outer radius a (*IV Escape*); which of the two events occurs is determined by comparing the magnitude of the probability fluxes through the two subdomain boundaries at time τ_r . If the event was an *IV Reaction* we directly replace the two particles by a particle of the product species at the new position of the center-of-motion \mathbf{R} ; if, instead, an *IV Escape* occurred, we know that $|\mathbf{r}| = a$ and still have to sample a new angle φ to construct the new interparticle vector \mathbf{r} . Finally, we transform the coordinates \mathbf{R} and \mathbf{r} back to new particle positions \mathbf{r}_A and \mathbf{r}_B . The detailed procedure of sampling the event type and new positions is described in supplementary sec. S1.2.2.

2.2.3 Binding to cylinders

We handle the binding of bulk particles to reactive cylindrical structures via the *Cylindrical Surface Interaction* domain, shown in Fig. 3A. Since only the radial distance of the particle from the cylinder is relevant to the binding problem, the natural geometry of this problem is again cylindrical, only that the cylindrical domain now has four boundaries: an inner boundary that wraps around the reactive cylinder at contact radius $\sigma = R_{\text{part}} + R_{\text{cyl}}$, an opposite absorbing outer boundary at a distant radius a , and two absorbing boundaries in axial (z) direction. As before, we can separate the polar and axial movement and determine two next-event times τ_r and τ_z , the smaller of which determines which coordinate hit the corresponding boundary first.

In the two polar coordinates, cylinder binding is akin to the two-particle reaction-diffusion problem formulated in the interparticle coordinate \mathbf{r} in sec. 2.2.2, and indeed we employ the same Green’s function $p_r^{\text{RA}}(r, \varphi, t|r_0, \varphi_0)$, obtained by solving the problem defined by Eqs. (8)–(11), to sample a next-event time τ_r for the polar motion; for the movement in z -direction, we use the Green’s function for one-dimensional free diffusion with two absorbing boundaries, which is a special case (for $v = 0$) of Green’s function p_X^{A} , introduced in the following section 2.2.4. As before, in the case $\tau_r > \tau_z$, i.e., when the particle first hits one of the radial boundaries, we compare the probability fluxes at the opposite boundaries of the radial coordinate to determine whether the event is a binding reaction to the cylinder (*IV Interaction*) or exit through the distant boundary (*IV Escape*); upon binding, the particle is placed onto the axis of the cylinder. In the case $\tau_z > \tau_r$, i.e. when the particle exits the cylindrical domain through one of its caps, the respective probability fluxes through the corresponding absorbing boundaries are compared to determine through which z -boundary the particle exits; the latter can be omitted when a symmetric cylinder centered at the initial particle position is used—then the boundary of exit can be chosen randomly with probability 1/2. In either case ($\tau_r > \tau_z$ or $\tau_z > \tau_r$) the new value of the respective other coordinate(s) is sampled from the corresponding Green’s function evaluated at the next-event time.

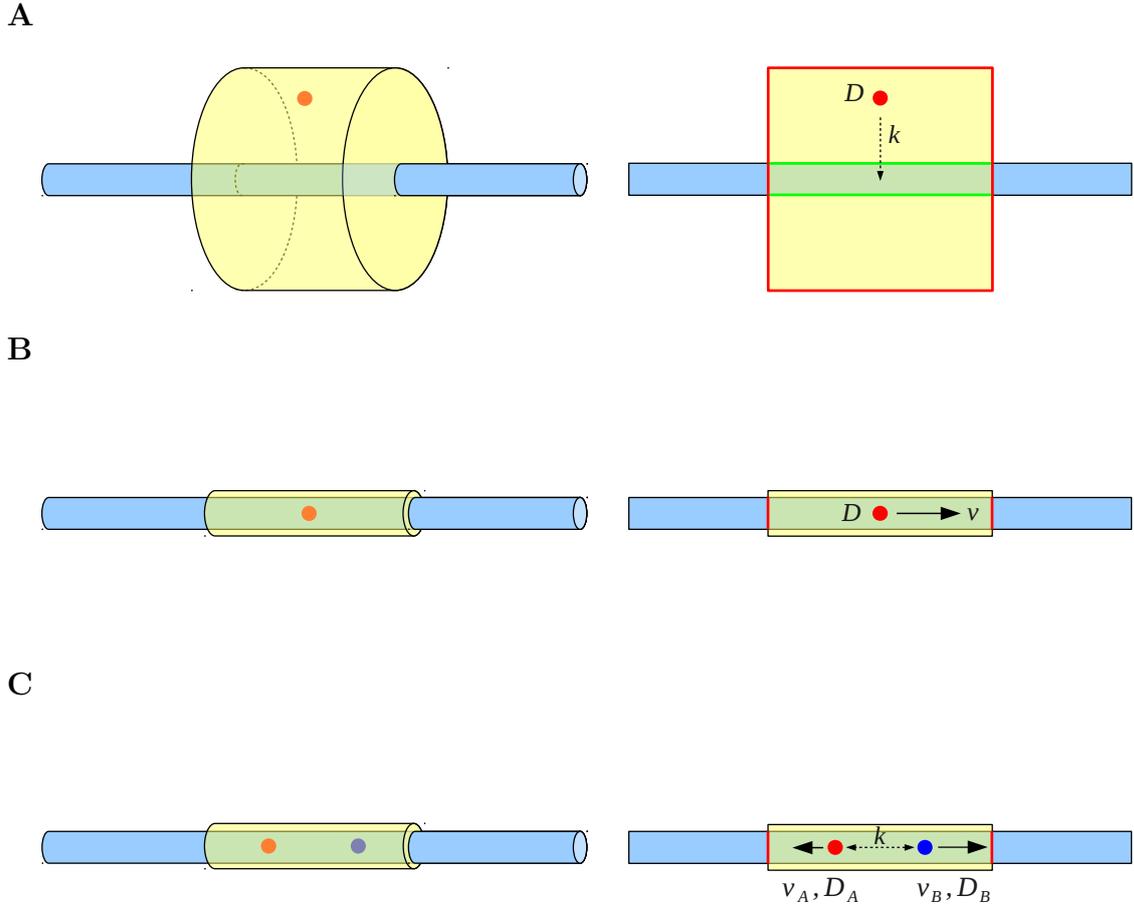


Figure 3: New protective domain types for interactions with and transport and reactions on 1D structures. (A) *Cylindrical Surface Interaction* domain; (B) *Cylindrical Surface Single* domain; (C) *Cylindrical Surface Pair* domain. Right panels show sections of 3D objects along the common cylinder axis. Absorbing boundaries are highlighted by red, radiating boundaries by green. Note that drift velocities (v , v_A , v_B) can be towards any cylinder end.

2.2.4 Movement and reactions on cylinders

In biological cells, motion confined to one-dimensional structures is widespread. Perhaps the most prominent examples are the diffusive search, hopping and sliding of transcription factors on the DNA [30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40], and active transport of proteins and vesicles by motor proteins on cytoskeletal filaments, such as microtubules and actin filaments [82, 83, 84, 85, 86, 87, 88, 89, 90]. In the latter case, processive motor proteins, such as members of the dynein and kinesin families, undergo an ATP-fueled periodic cycle of reactions, this way breaking detailed balance and creating a random walk with a clear directional bias on the filament, markedly different from simple diffusion.

In our new eGFRD implementation, particles that are bound to cylinders can move both via 1D diffusion and/or active transport, and engage in interparticle reactions. We model active transport by supplementing the PDE governing the time evolution of the PDF $p_x(x, t|x_0)$, describing the probability to find the particle at position x at

time t given an initial position x_0 , by an additional convection term with a constant “drift” velocity v , representative of the unidirectional motion component of active transport:

$$\partial_t p_x(x, t|x_0) = [D\partial_x^2 - v\partial_x] p_x(x, t|x_0) \quad (12)$$

This equation, together with the respective initial and boundary conditions, defines the boundary value problems that yield the Green’s functions for the *Cylindrical Surface Single* and *Cylindrical Surface Pair* domains, shown in Fig. 3B and 3C, used, respectively, to simulate the diffusion-drift process of a single particle on the cylinder and the interaction of two particles moving by diffusion and drift. Naturally, both domains are cylindrical, with a radius corresponding to the (largest) particle radius, and a length depending on the available free space in the surroundings of the cylinder-bound particle. The calculation of the necessary Green’s functions and the procedure of sampling the next-event time and type, and new positions of the particle(s), is analogous to the *Planar Surface Pair*, with minor additional precautions owed to the presence of the drift, as described further below. The Green’s function $p_x^A(x, t|x_0)$ used by the *Cylindrical Surface Single* is obtained by solving Eq. (12) subject to absorbing boundary conditions on both sides of the domain.

For the *Cylindrical Surface Pair*, the original equation governing the time-evolution of the two-particle PDF, $p_{2,x}(x_A, x_B, t|x_{0,A}, x_{0,B})$, reads

$$\partial_t p_{2,x} = [D_A\partial_{x_A}^2 + D_B\partial_{x_B}^2 - v_A\partial_{x_A} - v_B\partial_{x_B}] p_{2,x} \quad (13)$$

where each of the two particles has its own diffusion constant D_A (D_B) and drift velocity v_A (v_B). In the Supplementary Information, sec. S3.4, we show that the same coordinate transform as introduced in sec. 2.1 can be carried out also with the convection terms, again separating the above equation into one for the center-of-motion X and one for the interparticle separation $x = x_B - x_A$, which both have the form of Eq. (12), with the following diffusion coefficients and drift velocities for the transformed coordinates:

$$D_x = D_A + D_B \quad , \quad v_x = v_B - v_A \quad , \quad (14)$$

$$D_X = \frac{D_A D_B}{D_A + D_B} \quad , \quad v_X = \frac{D_B v_A + D_A v_B}{D_A + D_B} \quad , \quad (15)$$

For the center-of-motion coordinate X , the Green’s function is identical to the one used in the *Cylindrical Surface Single*, i.e. $p_x^A(X, t|X_0)$. For the interparticle Green’s function $p_x^{\text{RA}}(x, t|x_0)$ once more a radiating boundary condition has to be imposed at particle contact, $x = \sigma = R_A + R_B$, in order to model interparticle reactions. Here it is important to take into account that the inclusion of convective motion also changes the definition of the probability flux at the boundary; the radiating boundary condition at contact with diffusion and drift therefore reads:

$$-D\partial_x p(x, t)|_{x=\sigma} + vp(\sigma, t) = -kp(\sigma, t) \quad (16)$$

$$\Leftrightarrow \quad \partial_x p(x, t)|_{x=\sigma} = \frac{v+k}{D}p(\sigma, t) \quad (17)$$

where k stands for the intrinsic reaction rate at contact which in 1D has the same unit as the drift velocity v . The minus sign on the right side of Eq. (16) reflects the flux direction within the chosen coordinate system, which at the inner boundary for the interparticle separation x , by convention, is negative with respect to the x -axis.

We present the full derivation of the Green’s function p_x^{RA} in sec. S4.1 of the Supplementary Information, obtaining Green’s function p_x^A as a special case in sec. S4.1.2.

2.2.5 Connected reactive structures and transitions between them

While isolated planar and cylindrical reactive structures already are well-suited to conceptually study their effect on biochemical reactions, a more faithful representation of *in vivo* conditions requires the possibility to create closed 3D compartments bounded by reactive surfaces, and let particles transit between them. eGFRD2 allows the creation of such compartments from interconnected orthogonal planes, while particles can diffuse and react across their connection seams via special “*Transition*” domains. Moreover, cylindrical surfaces can be connected to planes via an “interface disk” structure, allowing for transitions from cylinder to plane and vice versa; the same structure can be also used to “cap” a finite cylinder such that particles can accumulate at its end and unbind into the bulk. The different *Transition* domains and their working principle are explained in more detail in sections S5.1–S5.2 of the Supplementary Information.

2.2.6 Further domains for special applications

We devised two further domains for special applications: The *Mixed Pair 2D-3D* domain, in which a bulk particle diffusing in 3D can, upon contact, directly react with a plane-bound particle diffusing in 2D (see supplementary sec. S5.4), and the *Cylindrical Surface Sink* domain, in which a 1D particle moving on a cylinder can interact with a static reactive “sink” (binding site) while diffusing over it (see supplementary sec. S5.3); this makes it possible to model, for example, the binding of transcription factors to their promoter, as we showed recently [40]. While for the former we make use of a special coordinate transform that allows us to employ Green’s functions already implemented for simpler binding scenarios, for the latter we derived a new Green’s function with specialized boundary conditions; the detailed calculations are found in the respective sections of the Supplementary Information listed above.

2.3 Brownian Dynamics based on the Reaction Volume Method provides a fallback propagation mode

While at low particle densities eGFRD can be orders of magnitude more efficient than Brownian Dynamics, the process of sampling next-event times and new positions inside eGFRD domains is computationally expensive, such that the usage of Brownian Dynamics is advantageous again when the domain size becomes comparable to the particle radius. The crux of GFRD is to construct domains that do not overlap with each other—indeed, this is what turns GFRD into an exact algorithm. However, even at low densities it can happen that more than two particles come so close to each other that only very small non-overlapping domains could be constructed; this can also occur when one single particle comes close to static structures (planes or cylinders) with which it cannot react. In such situations, we resort to Brownian Dynamics when eGFRD domains larger than a predefined minimal size cannot be made any more.

Therefore, our eGFRD simulator comprises a fully-featured BD simulator capable of simulating all modes of particle motion, particle reactions and particle-surface interactions, and is equipped with a set of rules that makes it possible to seamlessly shuffle particles between the two simulator types.

To guarantee that interparticle and particle-surface reactions fulfill detailed balance, we devised a new BD algorithm based on the “reaction volume method” (rvm-BD); rvm-BD is akin in spirit to the Reaction Brownian Dynamics scheme [91], but more versatile. At its heart, the new algorithm introduces a “reaction volume” around each reactive object from which forward reactions can occur; detailed balance is maintained by placing the unbinding particle inside the reaction volume with a properly

rescaled rate directly derived from the detailed balance condition. We implemented the scheme such that particles that can potentially interact with each other are automatically grouped into special “*Multi*” domains, each of which constitutes an independent rvm-BD simulator instance with a specific optimal reaction volume size and time step jointly determined from the set of rates involved; this way the performance of the propagation in rvm-BD mode is enhanced. We give a detailed description of the scheme, including a step-by-step derivation of the detailed balance condition, in sec. S2 in the Supplementary Information.

2.4 Performance

In order to assess the performance of eGFRD2, we profiled our new eGFRD implementation for representative simulation scenarios (both with and without reactions involved) by recording the CPU time per real (simulated) time as a function of the particle number, always comparing to simulations in which only rvm-BD is used to propagate the particles as a reference. Since in the lower dimensions particle crowding builds up much faster than in 3d, the profiling was carried out for each dimension separately, in order to avoid that the lowest dimension becomes the limiting factor and obstructs the performance gains in higher dimensions. The detailed profiling protocol and results are described in sec. S7 of the Supplementary Information. In brief, the profiling results demonstrate that our new eGFRD implementation outperforms rvm-BD by up to 3 orders of magnitude for particle densities $\lesssim 100/\mu m^d$ ($\lesssim 1000/\mu m^3$ in 3d, which translates to $\lesssim \mu M$ concentrations). Here it should be emphasized that rvm-BD is a smart BD scheme which optimally adapts the choice of the reaction volumes and propagation time steps to the set of rates involved; this typically results in average time steps (more than) 3 orders of magnitude larger than classical time step settings ensuring sufficiently fine resolution of particle collisions, such as $\Delta t \sim 10^{-6} \sigma^2/D$, where σ^2 is the particle contact radius and D the (interparticle) diffusion constant; brute-force schemes based on such “safe” (but comparably naive) choices of the time step thus are outperformed by up to 6 orders of magnitude by eGFRD. Moreover, we expect significant performance gains from a currently ongoing code optimization, as detailed in the subsequent section 2.5.

2.5 Code availability

A fully-functional prototype code of eGFRD2 is available online at GitHub¹. Most parts of the code, especially core-functions such as the scheduler system, the reaction-networks implementation and basic geometric objects, are written in C++, while Python has been used for the more top-level routines. The open-source *boost:python* libraries are used as an interface between C++ and Python. As a benefit of this, our implementation offers a user-friendly Python interface which can be used for fast and easy scripting of simulations and associated measurement routines. It also contains a visualization module based on *VPython*, and a module that exports the simulator output into a format readable by *Paraview*. While the use of Python comes with many user- and developer-friendly conveniences, we also found that it inflicts a significant overhead of computational cost. In the future, we will present a more efficient, fully overhauled code-version which outsources all remaining simulator parts into C++, only retaining the scripting interface in Python, thus minimizing its overhead.

¹<https://github.com/gfrd/egfrd/tree/develop>

3 Results: Simulation of Pom1 gradient formation

In order to apply the newly implemented eGFRD2 framework to a real biological problem and illustrate its capabilities we sought to study a simple but nontrivial reaction mechanism within which spatial features and different modes of biochemical transport play a prominent role. The reaction mechanism underlying the formation of the intracellular Pom1 gradient in bacteria, introduced below, ideally fulfills these criteria.

3.1 The Pom1 gradient

Protein gradients play a crucial role in cell biology; they map protein concentration levels to the distance from the gradient source, creating positional cues for downstream targets. The establishment of local protein accumulations acting as gradient sources oftentimes involves the cytoskeleton and active transport [92, 93, 94, 95, 96, 97, 98, 99, 100]. A representative example is the Pom1 gradient: Pom1 is a strongly membrane-associated auto-kinase that marks the division site in elongated fission yeast cells via concentration gradients decreasing from the cell poles [101, 102, 103, 104, 105]; the required source-accumulations of membrane-bound Pom1 are established by microtubules that direct cytoplasmic Pom1 towards the opposite poles. Recent experiments revealed that the membrane-associated gradient is shaped via a phosphorylation-dephosphorylation cycle of Pom1 [106]: Pom1 has its highest affinity in its dephosphorylated form; after membrane binding, dephosphorylated Pom1 starts self-phosphorylating, successively reducing its own membrane affinity as it diffuses away from the cell tip. Upon reaching higher phosphorylation levels it is recycled to the cytoplasm. Dephosphorylation of Pom1 by the phosphatase Dis2 is catalyzed by the polarity marker protein Tea4, which is transported towards the cell tips via active transport on microtubules and itself accumulates at the cell tips.

While in [106] the basic principle of this intricate gradient formation mechanism was uncovered, some important details remained unknown, in particular where and how precisely Dis2 dephosphorylates cytoplasmic Pom1, and whether the autophosphorylation occurs in a intramolecular or intermolecular fashion (cis- vs. trans-autophosphorylation). While other kinases from the DYRK family, to which Pom1 belongs, have been shown to undergo cis-autophosphorylation [107, 108, 109], a more recent study [110] based on new experiments and an ODE model concluded that membrane-bound Pom1 phosphorylates in trans-fashion; the resulting feedback of (local) Pom1 concentration on the (local) phosphorylation activity anti-correlates the gradient amplitude and length scale, thus implementing a buffering mechanism that makes the Pom1 density away from the gradient origin insensitive to Pom1 abundance and the rate of (dephosphorylated) Pom1 delivery to the membrane [110]. It was also shown that the underlying multi-step trans-phosphorylation on the membrane effectively is equivalent to a nonlinear membrane desorption rate with close-to-quadratic dependence on the Pom1 concentration. Notwithstanding the benefit of such mechanism for buffering against different initial conditions between cells (extrinsic noise), it remains unclear how exactly it can be successfully implemented under the highly stochastic conditions encountered within each single yeast cell (intrinsic noise); in particular, the nonlinear trans-phosphorylation at the heart of the scheme could amplify local density inhomogeneities, and thus introduce additional fluctuations in the gradient profile, especially at low Pom1 abundance or delivery rate.

3.2 Model

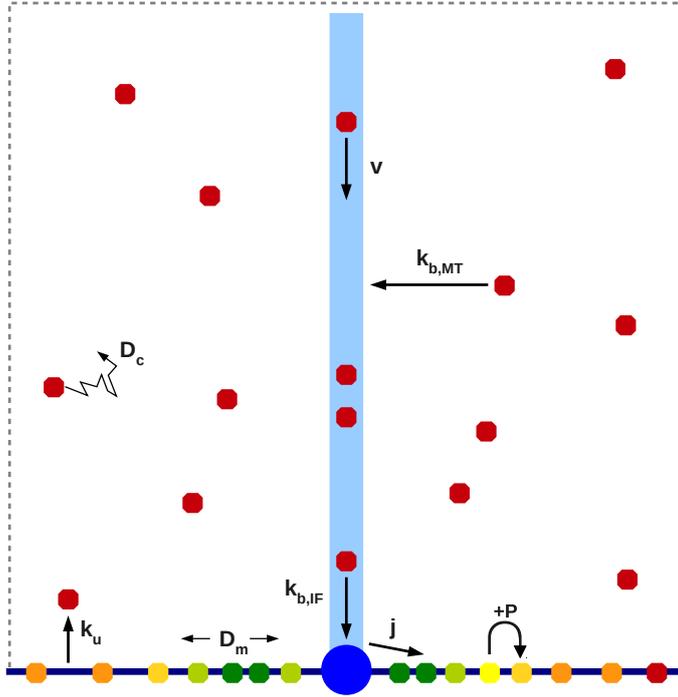
To address this question, we considered a stochastic particle-based model with a simplified geometry containing the minimal set of components necessary for Pom1 gradient formation on the membrane at one of the cell poles, shown in Fig. 4. The unfolded cell cortex is represented by a single (xy-) plane. A single static cylinder orthogonal to the plane (i.e. pointing in z-direction), representing a microtubule, intersects with the plane at its center.

As a further simplification, we do not explicitly include the Tea4/Dis2 dimers responsible for Pom1 dephosphorylation. Instead, we assume a static “conversion cluster” at the interface between membrane and microtubule, represented by a single static particle with a large radius. Dephosphorylation of Pom1 via Tea4/Dis2 is assumed to occur with Poissonian statistics for the complete conversion from the fully phosphorylated to the fully dephosphorylated state of Pom1.

Cytoplasmic Pom1 particles bind to the microtubule with a diffusion-limited rate. The microtubule-bound Pom1 then are actively transported towards the microtubule-membrane interface. There they are absorbed to the conversion cluster and unbind from it directly onto the membrane upon (complete) dephosphorylation. Membrane-bound Pom1 can autophosphorylate 6 times, as determined in experiments [106], thereby decreasing its affinity to the membrane; since it is unknown how exactly the Pom1 membrane unbinding rate increases with progressing phosphorylation level n , we decided for the arguably simplest choice, setting the unbinding rate to zero for $n \leq 5$ and to a finite value $k_u > 0$ for $n = 6$. We studied both cis- and trans-autophosphorylation, but mainly focused on the latter. Trans-phosphorylation is assumed to occur in a distributive fashion; however, in our simulations we find that fast rebindings tend to convert the distributive to a pseudo-processive scheme, as observed earlier in other systems [18]. While cis-autophosphorylation is a simple zero-order reaction process, the process of trans-autophosphorylation consists of three steps: first, the two involved Pom1 particles have to encounter each other via membrane diffusion and react to form a complex; secondly, the two particles bound in complex can carry out the actual phosphorylation of each other; finally, the complex dissociates into two monomers again. In our model, we make two simplifying assumptions: (1.) We assume that the rate of forming the complex is very high, such that the first (encounter) step is diffusion-limited; (2.) we treat the second and third steps as one process with Poissonian statistics described by a single “in-complex phosphorylation-dissociation rate” k_{pt} , assuming that the complex instantly dissociates after both monomers have increased their phosphorylation level by one; $\tau_{pt} \equiv 1/k_{pt}$ thus is the average waiting time between complex formation and dissociation into monomers with (one-fold) incremented phospholevels.

We used experimentally determined parameters if available, and reasonable estimates otherwise. Details of parameter choice are described in sec. S8.1 and Table S2 in the Supplementary Information.

A



B

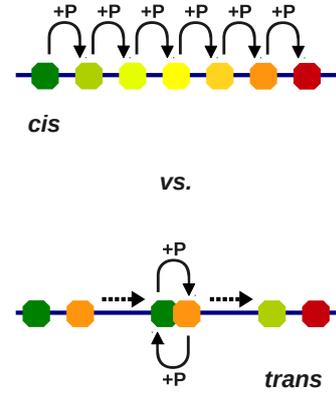


Figure 4: Pom1 gradient formation model. (A) Simplified 2D schematic of our Pom1 gradient formation model: A reactive plane (“membrane”, dark-blue line) is placed at the bottom of a 3D simulation box (grey-dotted lines) with uniform side length and periodic boundary conditions and connected to a reactive cylinder (“microtubule”, light-blue rectangle). Pom1 molecules (hexagons) can exist in 7 different phosphoforms ($n = 0$ / unphosphorylated to $n = 6$ / 6-fold phosphorylated, indicated by different colors); they diffuse in the cytoplasm with diffusion constant D_c and bind to the cylinder with diffusion-limited rate (reaction rate $k_{b,MT}$ at contact is very high). Cylinder-bound Pom1 undergoes a biased random walk towards the plane with drift velocity v and binds to the cylinder-plane interface with diffusion-limited rate (very high $k_{b,IF}$ at contact). The interface can absorb $N \geq 1$ Pom1 particles from the cylinder into a “conversion cluster” (blue circle) assumed to be responsible for Pom1 dephosphorylation. From the conversion cluster, single fully dephosphorylated Pom1 particles are unbound directly onto the membrane at random angle with injection rate j , where they diffuse with diffusion constant D_m and undergo successive autophosphorylation, using either the *cis* or *trans* scheme (explained below). The fully phosphorylated Pom1 ($n = 6$) can unbind back into the cytoplasm with rate k_u , thus closing the reaction cycle. (B) We consider two alternative autophosphorylation schemes: In *cis*-autophosphorylation (top) each particle can spontaneously self-phosphorylate towards the respective higher phosphoform with rate $k_{p,n}$. In the *trans*-scheme (bottom) two Pom1 particles first have to meet via diffusion and form a complex, within which they phosphorylate each other and dissociate again into two separate particles with increased phosphorylation level. For simplicity, here we assume that during this process the phosphorylation level of both particles is always incremented by one, and that the complex dissociates instantly afterwards; this combined autophosphorylation-dissociation process is governed by the same zero-order “in-complex phosphorylation-dissociation rate” k_{pt} for all combinations of phosphoforms. Moreover, we assume that the process of complex formation is diffusion-limited, meaning that the complex formation rate at contact is very high.

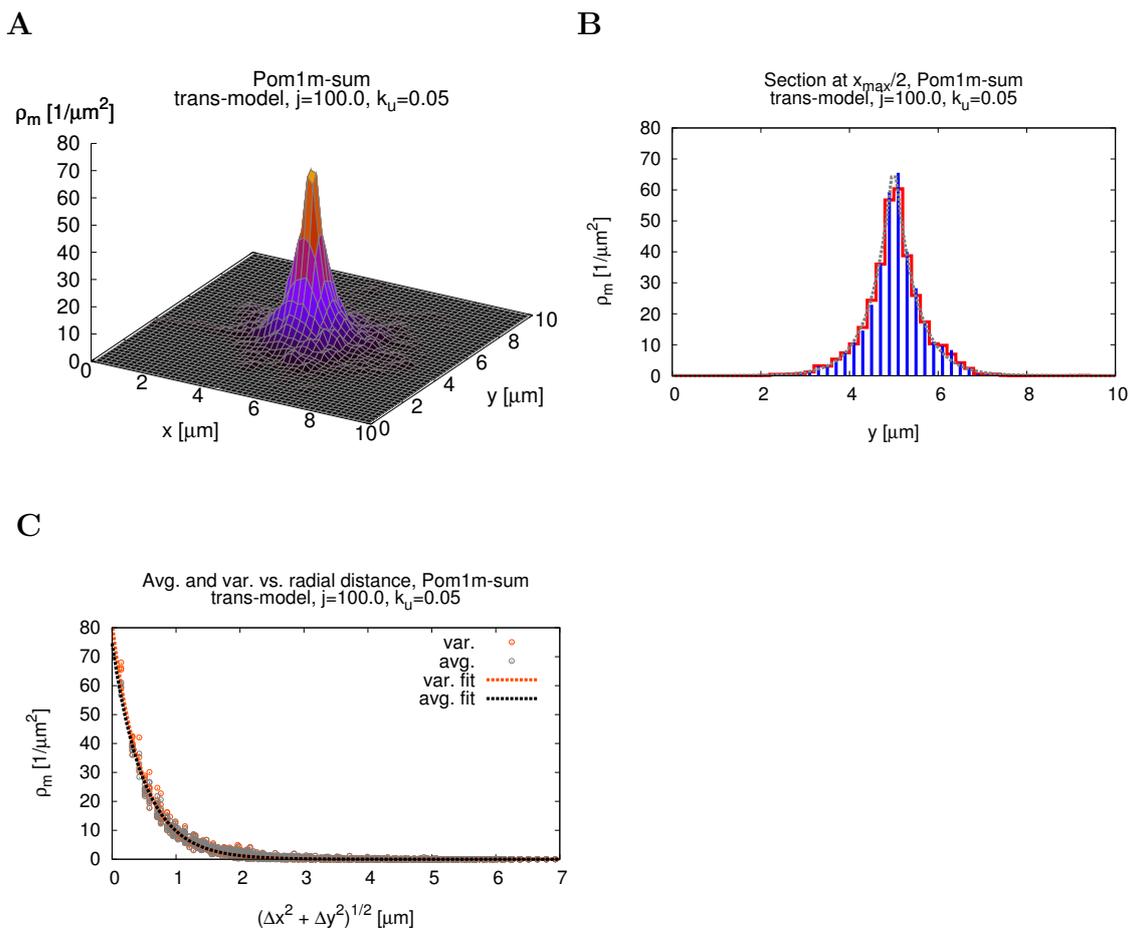


Figure 5: Example gradient (trans-model). (A) Histogram of total membrane-bound Pom1 density (comprising all phosphorylation states) on the membrane plane. (B) Section through histogram in (A) at $x = 5 \mu\text{m}$; red steps show the local density, blue impulses its variance, grey lines exponential fits to the data. (C) The same data plotted against the radial distance from the gradient origin, with exponential fits to the local mean density and its variance. Note that at all positions the variance essentially is in the Poisson limit.

3.3 Simulation results

We performed particle-based stochastic simulations of the model, using eGFRD extended by the new functionalities introduced in section 2. In order to be able to observe the buffering effect we varied—at constant particle number—the key parameters affecting the gradient amplitude and length scale: j , the “membrane injection rate” (rate of unbinding from the microtubule-membrane interface towards the membrane), and k_u , the rate of unbinding of the fully phosphorylated Pom1. Starting from random initial conditions with cytoplasmic, unphosphorylated Pom1 only, for each parameter set we first propagated the system for (at least) 30 s of simulated time, in order to allow it to form and equilibrate the gradient; we then recorded particle positions and species with a fixed data acquisition interval (0.1 s) for (at least) 30 s more of simulated time. Subsequently, we binned the local number of membrane-bound Pom1 species in a 2D histogram and thus obtained the stationary gradient profiles and its

local variance, which were further characterized by fitting exponential functions to the data. The details of the protocol are explained in sec. S8 of the Supplementary Information.

In Fig. 5 we show typical output from our simulations for a representative set of parameters (trans-model, $j = 100/s$, $k_u = 0.05/s$). Fig. 5A displays a 2D histogram of the total (membrane-bound) Pom1m density (summing all phosphoforms) on the plane, Fig. 5B is a y -section through the 3D profile at $x = 5 \mu m$; the variance of the local mean density in each histogram bin is plotted using blue impulses. In Fig. 5C we present the same data as a function of the radial distance from the gradient origin (at $x = 5 \mu m$, $y = 5 \mu m$) in a scatter plot (local mean density = grey circles, local density variance = orange circles) and the exponential functions fitted to both the density and variance data. Figs. 5B and 5C demonstrate that (1.) the shape of the stationary gradient profile is well approximated by an exponential function (as already observed in [110]), and (2.) that the local density fluctuations are essentially in the Poissonian limit (variance \simeq mean). These findings are representative for all parameter sets that we present in the following.

Fig. 6 shows a more systematic analysis of the key gradient properties, comparing different parameter sets. In Fig. 6A and Fig. 6B we plot the fitted gradient amplitude A and length scale L , respectively, as a function of the injection rate j . Not surprisingly, the amplitude A is an increasing function of the Pom1 injection rate at the gradient origin, but saturates for $j \gtrsim 30/s$ when the overall abundance of Pom1 in the system and the flux of microtubule-bound Pom1 towards the membrane become the limiting factors; this holds for all considered unbinding rates k_u , but the maximal amplitude slightly reduces with increasing k_u . In contrast, the gradient length scale decreases with increasing Pom1 injection at the origin; this is precisely the effect of enhancing the effective unbinding rate and thus shortening the gradient via trans-autophosphorylation and its nonlinear dependence on the Pom1 concentration. For the faster unbinding rates $k_u \geq 0.5/s$ the gradient length scale L is essentially independent of k_u , because in this regime only the average time required for full phosphorylation matters; remarkably, L becomes insensitive to further increases of k_u much earlier than the amplitude A .

In Fig. 6C we plot the fitted length scale against the fitted amplitude on a double-logarithmic scale. The figure shows that gradient length and amplitude indeed are anti-correlated, but the degree of anticorrelation depends on the unbinding rate, as demonstrated by the fitted slopes: for slow unbinding ($k_u = 0.05/s$), the amplitude decreases less strongly with increasing length scale (slope $m = -0.19$), while the theoretically predicted [110] slope $m = -0.5$ (grey line) is only approached for the faster unbinding rates ($k_u \geq 5/s$).

Fig. 6D summarizes the noise properties of the gradients. Here we plot the local variance (density) against the local mean density in a scatter plot, whereby each point corresponds to a single bin of a two-dimensional histogram as shown in Fig. 4A, for all parameter sets considered. The colors correspond to the ones used in Figs. 6A-6C and distinguish points from parameter sets with a certain value of the unbinding rate k_u ; the point clouds of one particular color thus include values for different positions on the plane and different injection rates j . The plot shows that the gradient profiles for all parameter values considered are, in essence, in the Poissonian limit (variance = mean, grey line in the plot), apart from a small deviation towards slightly higher variances at the highest Pom1 densities, corresponding to low unbinding rate ($k_u = 0.05/s$) and positions close to the gradient origin, where unphosphorylated Pom1 is injected onto the membrane.

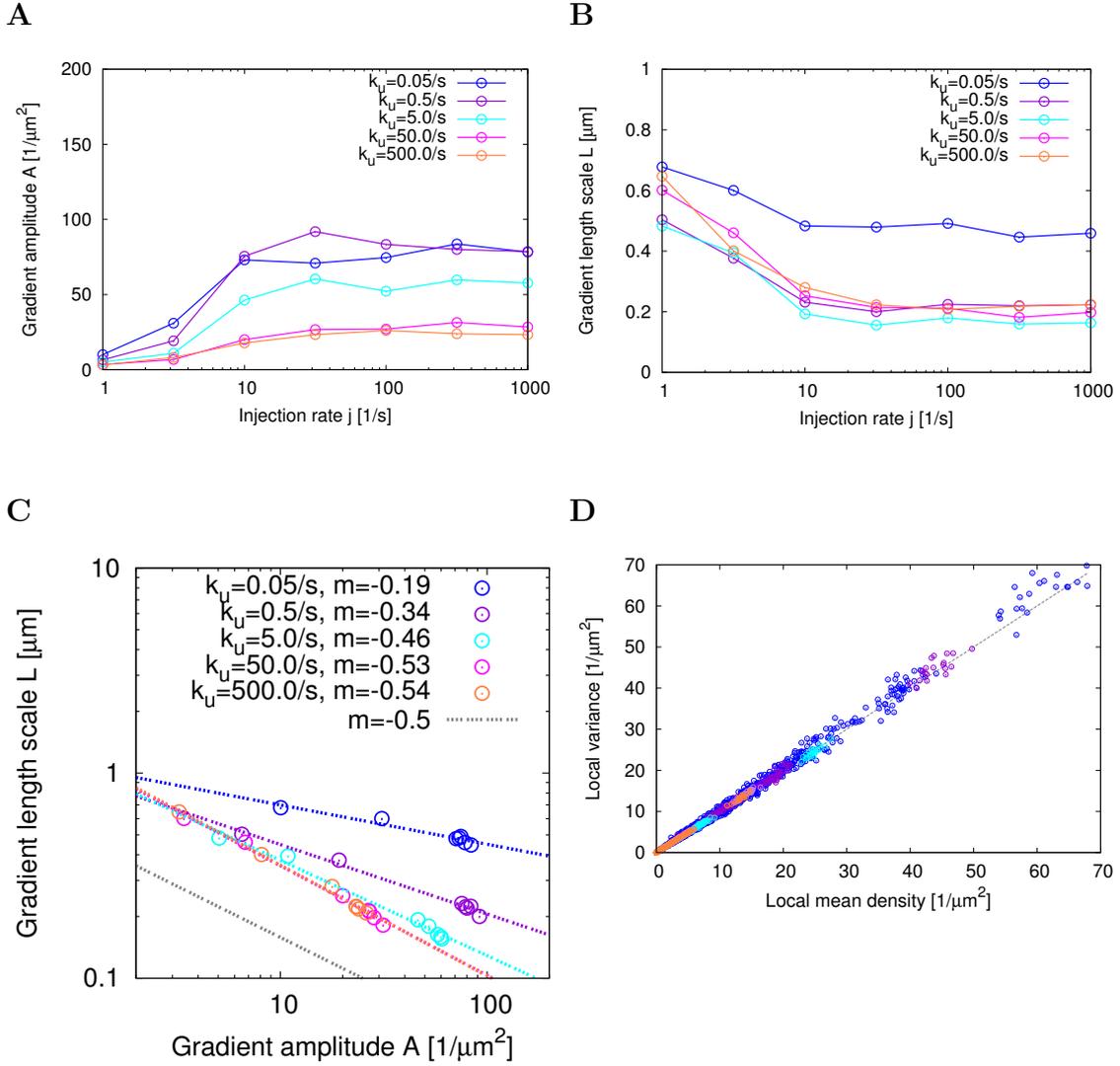


Figure 6: Gradient characteristics as a function of key parameters (trans-model). Gradient amplitude (A) and length scale L (B), obtained by fitting an exponential function to stationary radial profiles (cf. Fig. 5C), plotted against the injection rate j at the gradient origin for different unbinding rates k_u of the fully phosphorylated Pom1. Lines are guidelines for the eye. Note that L becomes shorter for increasing j due to increased concentration at the membrane, resulting in higher probability of autophosphorylation and faster unbinding to the cytoplasm. (C) Log-log plot of the gradient length L vs. gradient amplitude A for the same data. The grey line shows the theoretically predicted [110] scaling (slope $m = -0.5$), only reached in the limit of fast unbinding of fully phosphorylated Pom1. (D) Variance vs. mean for the local total membrane-bound Pom1 density, combining the values of all histogram bins into one scatter plot; point colors correspond to the colors in (A) - (C) (different values of k_u). The data shown was obtained in the limit of very fast in-complex phosphorylation-dissociation rate ($k_{pt} = 1000/s$), such that the overall rate of autophosphorylation is (membrane-) diffusion limited.

3.4 Influence of the trans-phosphorylation rate

In order to test how much our results depend on our choice of very fast in-complex phosphorylation-dissociation rate ($k_{\text{pt}} = 1000/s$), we varied this parameter at a constant, intermediately fast unbinding rate $k_{\text{u}} = 5/s$. Here we only summarize the principal observations and comment on them in more detail in sec. S8.3 of the Supplementary Information. We observe that while the gradient amplitude markedly increases with increasing rate k_{pt} , the scaling of the anticorrelation between gradient length and amplitude is preserved as long as $k_{\text{pt}} \gtrsim 10/s$. Moreover, for all values of k_{pt} , the noise remains Poissonian. The buffering effect observed in the trans-model therefore turns out to be very robust with respect to k_{pt} , the rate governing autophosphorylation in complex and subsequent dissociation.

3.5 The cis-phosphorylation model

In order to compare the trans- and cis-autophosphorylation schemes, we repeated the simulations above also for the cis-scheme. The results are presented in Fig. S10 in the Supplementary Information and only briefly summarized here. In accordance to previous results and our initial expectations, the cis-scheme does not provide any buffering mechanism that compensates increases in amplitude with decreases in length scale. However, both the amplitude and length scale are remarkably invariant to most of the relevant system parameters, in particular to the membrane unbinding rate k_{u} , highlighting that the successive Poissonian phosphorylation steps of the cis-scheme effectively provide a “timer” function for the unbinding from the membrane.

4 Conclusion and Outlook

Due to limited tractability of analytical models, and thanks to advances both in biophysical theory and computational power during the last decades, spatial-stochastic simulations have become an important tool for exploring the mechanistic behavior of complex biochemical systems. Driven by the continuous desire to make simulations more detailed and realistic, and by the recognition of the fact that in biological cells a myriad of chemical species coexists at predominantly low copy numbers, particle-based simulation techniques, mostly based on the principle of Brownian Dynamics, recently have taken a prominent role among these efforts. While substantial achievements have been made in making Brownian Dynamics more realistic and biochemically accurate, even with the recent advances they remain computationally demanding; with time steps typically in the μs regime and below, the run time required to simulate 1 s of real time for a system of ~ 1000 particles can easily reach the order of $\sim 10^6$ s (~ 10 d).

To overcome these limitations, the community has recently begun to pursue two distinct approaches: the parallelization of particle-based simulation algorithms, in order to allow them to exploit the full power of large CPU or GPU clusters [68, 52, 111, 112, 113], and the development of hybrid techniques that shuffle particles between simulators with coarser or finer spatial resolution, depending on the local density [114, 115, 116, 117, 118, 119]. eGFRD puts forward a different approach. The key idea of eGFRD is to get rid of “unnecessary” detail, namely the microscopic trajectories of diffusive motion between particle encounters, while retaining all other, “informative” details, without sacrificing access to individual particle-positions at any time, and without compromising accuracy. To this end, eGFRD partitions the original N -particle system into $M < N$ smaller, analytically tractable systems for which an exact solution (probability distribution) for the time evolution of the underlying stochastic

dynamics can be obtained. This allows to implement an event-driven algorithm in which the subsystems are updated locally and asynchronously, while the overall behavior of the full system is correctly preserved, and individual particle positions can be sampled with exact statistics at any desired time.

While this approach endows eGFRD with an extraordinarily high computational efficiency, reducing the CPU time per simulated time up to 10^6 -fold compared to brute-force Brownian Dynamics, until now eGFRD has been limited to a classical three-dimensional setting, such that an adequate representation of the intracellular architecture, oftentimes featuring lower-dimensional structures like membranes and elongated macropolymers, was not possible. In this work, we presented eGFRD2, a new version of eGFRD which incorporates finite low-dimensional reactive structures, and derived the Green's functions describing the reaction-diffusion processes of particles on these structures, and their interactions with them. Using these exact analytical solutions, we implemented a diverse set of new protective domains that seamlessly integrates with the spherical protective domains of the original eGFRD, and supplemented the new eGFRD with an efficient and accurate Brownian Dynamics scheme (rvm-BD) capable of propagating particles in all dimensions with situation-adaptive time steps. A benchmarking carried out for biologically representative parameters reveals that the eGFRD2 simulator—for sufficiently low particle densities—is up to 3 orders of magnitude more efficient than rvm-BD, and even more so compared to more naive BD schemes; however, in the lower dimensions the crossover point at which BD becomes more efficient than eGFRD is approached faster, because the effects of crowding build up more quickly as the dimension decreases. While our framework is very sophisticated and efficient in computing next-event times and new particle positions, the methods used for domain (re)creation are still comparably simplistic; yet, these processes make up a considerable amount of the computational effort required, especially when the simulation space gets more crowded. Using more sophisticated domain making schemes therefore is expected to further improve the performance of eGFRD2. Moreover, we find that the use of hybrid code (C++ for core routines, Python for more upstream routines and the user interface) comes at the cost of sacrificing efficiency. Forthcoming releases of our simulator will reduce the amount of Python code to a minimum required for user-friendliness, and is expected to result in significant further increases in simulation efficiency. In fact, as a first step we have recently rewritten the eGFRD code for simulating 3D systems, such that all routines are now in C++. This code is up to 6 orders of magnitude faster than brute-force BD².

As an example application, we used the new eGFRD2 framework to carry out particle-based simulations of Pom1 gradient formation; this process is driven by a reaction cycle in which fully phosphorylated Pom1 is collected from the cytoplasm by a microtubule, afterwards directed towards the membrane via active transport, and injected onto the membrane in its fully dephosphorylated state; on the membrane, the gradient then is shaped by diffusion and a multi-step phosphorylation cascade tuning the Pom1 unbinding rate. Comparing a trans-autophosphorylation mechanism to cis-autophosphorylation, we varied the crucial parameters of a minimal model capturing all essential processes involved in Pom1 gradient formation, and recorded how they affect the stationary gradient profiles. Our results confirm the buffering effect arising from an anticorrelation between the gradient amplitude and length scale in the trans-phosphorylation model, found by earlier studies [106, 110]; in addition, we find that even at low copy numbers the fluctuations in the gradient concentration are Poissonian at any distance from the source. Our results also suggest an important role for the trans-phosphorylation rate: the predicted scaling is only achieved for sufficiently

²See: gfrd.org

fast trans-phosphorylation and complex dissociation. Finally, while—as expected—no similar buffering effect can be observed in the cis-phosphorylation model, we find that here the emerging gradients are remarkably insensitive to most of the varied parameters, highlighting diametral benefits of the two different autophosphorylation schemes (buffering or “elasticity” in the trans-model vs. insusceptibility in the cis-model).

Our ongoing efforts pursue three directions: (1.) Since the tractability of Green’s function derivations demands working with simple, abstracted geometries, the level of detail with which real cell environments can be represented in eGFRD remains limited. However, since eGFRD necessarily must be integrated with a BD fallback simulator, this offers the opportunity to resort to BD simulations on triangulated structures if desired. We currently work on integrating particle motion and reactions on triangulated meshes with eGFRD. (2.) Also the fact that all particles are treated as spheres with uniform surface reactivity limits the level of detail in eGFRD. In reality, large molecules in particular have a markedly non-spherical structure with well-defined reactive spots, and upon coming close first have to engage in rotational diffusion within interaction potentials before being able to form a complex. In order to equip GFRD with the capability of resolving the reaction process with such detail, recently some of us developed MD-GFRD (“Molecular Dynamics GFRD”) [120, 121], which allows to propagate spherical particles with well-defined reactive patches via LANGEVIN dynamics once they come close together, with the option to switch to molecular dynamics at even closer particle proximity if desired. This principle will be fully integrated into eGFRD in the future. (3.) While the parallelization of event-driven spatial simulators is a daunting task, because different parts of the simulated space may quickly desynchronize, it has been recently achieved for a simple (3d only) eGFRD variant, named pGFRD, as part of the e-cell project [122]. Future versions of our eGFRD implementation will borrow from the techniques developed in pGFRD. Until these further extensions of eGFRD are fully elaborated, the framework presented here provides the community with an efficient quantitative tool for studying the behavior of biochemical systems in geometries representing the essential architecture of real cells, perfectly suited to explore the impact of principal geometric constraints on biochemical reactions at low concentrations in a particle-based, genuinely stochastic setting.

5 Acknowledgements

This work is part of the research programme of the Netherlands Organisation for Scientific Research (NWO) and was performed at the research institute AMOLF.

The authors thank Filipe Tostevin, Wiet H. de Ronde, Thomas E. Ouldridge, Andrew Mugler and Sorin Tănase-Nicola for many useful discussions about the principles that eGFRD is based on.

References

- [1] Yu J, Xiao J, Ren X, Lao K, Xie XS (2006) Probing Gene Expression in Live Cells, One Protein Molecule at a Time. *Science* **311**:1600.
- [2] Elf J, Li GW, Xie XS (2007) Probing Transcription Factor Dynamics at the Single-Molecule Level in a Living Cell. *Science* **316**:1191.
- [3] Simicevic J, Schmid AW, Gilardoni PA, Zoller B, Raghav SK, Krier I, Gubelmann C, Lisacek F, Naef F, Moniatte M, Deplancke B (2013) Absolute quantification of transcription factors during cellular differentiation using multiplexed targeted proteomics. *Nat Meth* **10**:570–576.
- [4] Li GW, Burkhardt D, Gross C, Weissman J (2014) Quantifying Absolute Protein Synthesis Rates Reveals Principles Underlying Allocation of Cellular Resources. *Cell* **157**:624–635.
- [5] Garza de Leon F, Sellars L, Stracy M, Busby SJW, Kapanidis AN (2017) Tracking Low-Copy Transcription Factors in Living Bacteria: The Case of the lac Repressor. *Biophys J* **112**:1316–1327.
- [6] Zimmerman SB, Minton AP (1993) Macromolecular crowding: biochemical, biophysical, and physiological consequences. *Annu Rev Biophys Biomol Struct* **22**:27–65.
- [7] Ellis RJ (2001) Macromolecular crowding: an important but neglected aspect of the intracellular environment. *Curr Opin Struct Biol* **11**:114–119.
- [8] Ellis RJ (2001) Macromolecular crowding: obvious but underappreciated. *Trends Biochem Sci* **26**:597–604.
- [9] Li GW, Berg OG, Elf J (2009) Effects of macromolecular crowding and DNA looping on gene regulation kinetics. *Nat Phys* **5**:294–297.
- [10] Zhou HX (2013) Influence of crowded cellular environments on protein folding, binding, and oligomerization: Biological consequences and potentials of atomistic modeling. *FEBS Letters* **587**:1053–1061.
- [11] Długosz M, Trylska J (2011) Diffusion in crowded biological environments: applications of Brownian dynamics. *BMC Biophys* **4**:3.
- [12] Mahmutovic A, Fange D, Berg OG, Elf J (2012) Lost in presumption: stochastic reactions in spatial models. *Nat Meth* **9**:1163–1166.
- [13] Tsimring LS (2014) Noise in biology. *Rep Prog Phys* **77**:026601.
- [14] Elf J, Ehrenberg M (2004) Spontaneous separation of bi-stable biochemical systems into spatial domains of opposite phases. *Syst Biol (Stevenage)* **1**:230–236.
- [15] Lawson MJ, Petzold L, Hellander A (2015) Accuracy of the Michaelis–Menten approximation when analysing effects of molecular noise. *J R Soc Interface* **12**:20150054.
- [16] van Albada SB, ten Wolde PR (2007) Enzyme Localization Can Drastically Affect Signal Amplification in Signal Transduction Pathways. *PLoS Comput Biol* **3**:e195.
- [17] Morelli MJ, ten Wolde PR (2008) Reaction Brownian dynamics and the effect of spatial fluctuations on the gain of a push-pull network. *J Chem Phys* **129**:054112 1–11.
- [18] Takahashi K, Tănase-Nicola S, ten Wolde PR (2010) Spatio-temporal correlations can drastically change the response of a MAPK pathway. *Proc Natl Acad Sci USA* **107**:2473–2478.

- [19] Dushek O, van der Merwe PA, Shahrezaei V (2011) Ultrasensitivity in Multisite Phosphorylation of Membrane-Anchored Proteins. *Biophys J* **100**:1189–1197.
- [20] Jilkine A, Angenent SB, Wu LF, Altschuler SJ (2011) A Density-Dependent Switch Drives Stochastic Clustering and Polarization of Signaling Molecules. *PLoS Comput Biol* **7**:e1002271.
- [21] Mugler A, Tostevin F, ten Wolde PR (2013) Spatial partitioning improves the reliability of biochemical signaling. *Proc Natl Acad Sci USA* **110**:5927–5932.
- [22] Wehrens M, ten Wolde PR, Mugler A (2014) Positive feedback can lead to dynamic nanometer-scale clustering on cell membranes. *J Chem Phys* **141**:205102.
- [23] Morelli M, Allen R, ten Wolde PR (2011) Effects of Macromolecular Crowding on Genetic Networks. *Biophys J* **101**:2882–2891.
- [24] Klann MT, Lapin A, Reuss M (2009) Stochastic Simulation of Signal Transduction: Impact of the Cellular Architecture on Diffusion. *Biophys J* **96**:5122–5129.
- [25] Klann MT, Lapin A, Reuss M (2011) Agent-based simulation of reactions in the crowded and structured intracellular environment: Influence of mobility and location of the reactants. *BMC Syst Biol* **5**:71.
- [26] ten Wolde PR, Mugler A (2014) Chapter Twelve - Importance of Crowding in Signaling, Genetic, and Metabolic Networks. In: Jeon RHaKW, editor, *Int Rev Cell Mol Biol*, Academic Press, volume 307 of *New Models of the Cell Nucleus: Crowding, Entropic Forces, Phase Separation, and Fractals*. pp. 419–442.
- [27] van Zon JS, Morelli MJ, Tănase-Nicola S, ten Wolde PR (2006) Diffusion of Transcription Factors Can Drastically Enhance the Noise in Gene Expression. *Biophys J* **91**:4350 - 4367.
- [28] Mugler A, Gotway Bailey A, Takahashi K, ten Wolde PR (2012) Membrane clustering and the role of rebinding in biochemical signaling. *Biophys J* **102**:1069–1078.
- [29] Kaizu K, de Ronde W, Paijmans J, Takahashi K, Tostevin F, ten Wolde PR (2014) The Berg-Purcell Limit Revisited. *Biophys J* **106**:976–985.
- [30] Wang YM, Austin RH, Cox EC (2006) Single Molecule Measurements of Repressor Protein 1D Diffusion on DNA. *Phys Rev Lett* **97**:048302.
- [31] Bonnet I, Biebricher A, Pierre-Louis P, Loverdo C, Bénichou O, Voituriez R, Escudé C, Wende W, Pingoud A, Desbiolles P (2008) Sliding and jumping of single EcoRV restriction enzymes on non-cognate DNA. *Nucl Acids Res* **36**:4118–4127.
- [32] Tafvizi A, Huang F, Fersht AR, Mirny LA, van Oijen AM (2011) A single-molecule characterization of p53 search on DNA. *Proc Natl Acad Sci U S A* **108**:563–568.
- [33] Hammar P, Leroy P, Mahmutovic A, Marklund EG, Berg OG, Elf J (2012) The lac Repressor Displays Facilitated Diffusion in Living Cells. *Science* **336**:1595–1598.
- [34] Nguyen B, Sokoloski J, Galletto R, Elson EL, Wold MS, Lohman TM (2014) Diffusion of Human Replication Protein A along Single-Stranded DNA. *J Mol Biol* **426**:3246–3261.
- [35] Loverdo C, Bénichou O, Moreau M, Voituriez R (2008) Enhanced reaction kinetics in biological cells. *Nat Phys* **4**:134–137.
- [36] Loverdo C, Bénichou O, Voituriez R, Biebricher A, Bonnet I, Desbiolles P (2009) Quantifying Hopping and Jumping in Facilitated Diffusion of DNA-Binding Proteins. *Phys Rev Lett* **102**:188101.

- [37] Loverdo C, Bénichou O, Moreau M, Voituriez R (2009) Robustness of optimal intermittent search strategies in one, two, and three dimensions. *Phys Rev E* **80**:031146.
- [38] Bénichou O, Chevalier C, Klafter J, Meyer B, Voituriez R (2010) Geometry-controlled kinetics. *Nat Chem* **2**:472–477.
- [39] Bénichou O, Loverdo C, Moreau M, Voituriez R (2011) Intermittent search strategies. *Rev Mod Phys* **83**:81–129.
- [40] Paijmans J, ten Wolde PR (2014) Lower bound on the precision of transcriptional regulation and why facilitated diffusion can reduce noise in gene expression. *Phys Rev E* **90**:032708.
- [41] Schwarz K, Schröder Y, Qu B, Hoth M, Rieger H (2016) Optimality of Spatially Inhomogeneous Search Strategies. *Phys Rev Lett* **117**:068101.
- [42] Fange D, Elf J (2006) Noise-Induced Min Phenotypes in *E. coli*. *PLoS Comput Biol* **2**:e80.
- [43] Mugler A, ten Wolde PR (2013) The Macroscopic Effects of Microscopic Heterogeneity in Cell Signaling. In: Rice SA, Dinner AR, editors, *Adv Chem Phys*, John Wiley & Sons, Inc. pp. 373–396.
- [44] Dobrzyński M, Rodríguez JV, Kaandorp JA, Blom JG (2007) Computational methods for diffusion-influenced biochemical reactions. *Bioinformatics* **23**:1969–1977.
- [45] Sokolowski TR, ten Wolde PR (2017). Spatial-Stochastic Simulation of Reaction-Diffusion Systems. arXiv:1705.08669 [q-bio.MN].
- [46] Hattne J, Fange D, Elf J (2005) Stochastic reaction-diffusion simulation with MesoRD. *Bioinformatics* **21**:2923–2924.
- [47] Wang S, Elf J, Hellander S, Lötstedt P (2013) Stochastic Reaction-Diffusion Processes with Embedded Lower-Dimensional Structures. *Bull Math Biol* **76**:819–853.
- [48] Drawert B, Engblom S, Hellander A (2012) URDME: a modular framework for stochastic simulation of reaction-transport processes in complex geometries. *BMC Syst Biol* **6**:76.
- [49] Drawert B, Lawson MJ, Petzold L, Khammash M (2010) The diffusive finite state projection algorithm for efficient simulation of the stochastic reaction-diffusion master equation. *J Chem Phys* **132**:074101.
- [50] Lampoudi S, Gillespie DT, Petzold LR (2009) The multinomial simulation algorithm for discrete stochastic simulation of reaction-diffusion systems. *J Chem Phys* **130**:094104.
- [51] Isaacson S, Peskin C (2006) Incorporating Diffusion in Complex Geometries into Stochastic Chemical Kinetics Simulations. *SIAM J Sci Comput* **28**:47–74.
- [52] Drawert B, Hellander A, Bales B, Banerjee D, Bellesia G, Daigle BJ Jr, Douglas G, Gu M, Gupta A, Hellander S, Horuk C, Nath D, Takkar A, Wu S, Lötstedt P, Krintz C, Petzold LR (2016) Stochastic Simulation Service: Bridging the Gap between the Computational Expert and the Biologist. *PLoS Comput Biol* **12**:1-15.
- [53] Moraru II, Schaff JC, Slepchenko BM, Blinov ML, Morgan F, Lakshminarayana A, Gao F, Li Y, Loew LM (2008) Virtual Cell modelling and simulation software environment. *IET Syst Biol* **2**:352–362.

- [54] Rodríguez JV, Kaandorp JA, Dobrzyński M, Blom JG (2006) Spatial stochastic modelling of the phosphoenolpyruvate-dependent phosphotransferase (PTS) pathway in *Escherichia coli*. *Bioinformatics* **22**:1895–1901.
- [55] Vigelius M, Lane A, Meyer B (2010) Accelerating Reaction-Diffusion Simulations with General-Purpose Graphics Processing Units. *Bioinformatics* :btq622.
- [56] Gillespie D (1976) A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J Comput Phys* **22**:403–434.
- [57] Gillespie D (1977) Exact stochastic simulation of coupled chemical reactions. *J Chem Phys* **81**:2340–2361.
- [58] Andrews SS, Bray D (2004) Stochastic simulation of chemical reactions with spatial resolution and single molecule detail. *Phys Biol* **1**:137.
- [59] Andrews SS, Addy NJ, Brent R, Arkin AP (2010) Detailed Simulations of Cell Biology with Smoldyn 2.1. *PLoS Comput Biol* **6**:e1000705.
- [60] Stiles JR, van Helden D, Bartol TM, Salpeter EE, Salpeter MM (1996) Miniature endplate current rise times <100 microseconds from improved recordings can be modeled with passive acetylcholine diffusion from a synaptic vesicle. *Proc Natl Acad Sci USA* **93**:5747–5752.
- [61] Stiles JR, Bartol TM (2000) Monte Carlo Methods for Simulating Realistic Synaptic Microphysiology Using MCell. In: Computational Neuroscience, CRC Press, Frontiers in Neuroscience.
- [62] Franks KM, Bartol TM, Sejnowski TJ (2002) A Monte Carlo Model Reveals Independent Signaling at Central Glutamatergic Synapse neuromuscular junction. *Biophys J* **83**:2333–2348.
- [63] Kerr R, Bartol T, Kaminsky B, Dittrich M, Chang J, Baden S, Sejnowski T, Stiles J (2008) Fast Monte Carlo Simulation Methods for Biological Reaction-Diffusion Systems in Solution and on Surfaces. *SIAM J Sci Comput* **30**:3126–3149.
- [64] Stefan MI, Bartol TM, Sejnowski TJ, Kennedy MB (2014) Multi-state Modeling of Biomolecules. *PLoS Comput Biol* **10**:e1003844.
- [65] Plimpton SJ, Slepoy A (2005) Microbial cell modeling via reacting diffusing particles. *J Phys Conf Ser* **16**:305–309.
- [66] Boulianne L, Assaad SA, Dumontier M, Gross WJ (2008) GridCell: a stochastic particle-based biological system simulator. *BMC Syst Biol* **2**:66.
- [67] Arjunan SNV, Tomita M (2009) A new multicompartmental reaction-diffusion modeling method links transient membrane attachment of *E. coli*. *Syst Synth Biol* **4**:35–53.
- [68] Miyauchi A, Iwamoto K, Arjunan SNV, Takahashi K (2016) pSpatiocyte: A Parallel Stochastic Method for Particle Reaction-Diffusion Systems. *arXiv:160503726 [q-bio]* .
- [69] Schöneberg J, Noé F (2013) ReaDDy - A Software for Particle-Based Reaction-Diffusion Dynamics in Crowded Cellular Environments. *PLoS ONE* **8**:e74261.
- [70] van Zon JS, ten Wolde PR (2005) Simulating biochemical networks at the particle level and in time and space: Green’s function reaction dynamics. *Phys Rev Lett* **94**:128103 1–4.
- [71] van Zon JS, ten Wolde PR (2005) Green’s-function reaction dynamics: a particle-based approach for simulating biochemical networks in time and space. *J Chem Phys* **123**:234910 1–16.

- [72] Janmey PA (1998) The cytoskeleton and cell signaling: component localization and mechanical coupling. *Physiol Rev* **78**:763–781.
- [73] Kholodenko BN (2006) Cell-signalling dynamics in time and space. *Nat Rev Mol Cell Biol* **7**:165–176.
- [74] Moseley JB, Goode BL (2006) The Yeast Actin Cytoskeleton: from Cellular Function to Biochemical Mechanism. *Microbiol Mol Biol Rev* **70**:605–645.
- [75] Kuchler A, Yoshimoto M, Luginbuhl S, Mavelli F, Walde P (2016) Enzymatic reactions in confined environments. *Nat Nano* **11**:409–420.
- [76] Ooppelstrup T, Bulatov VV, Gilmer GH, Kalos MH, Sadigh B (2006) First-Passage Monte Carlo Algorithm: Diffusion without All the Hops. *Phys Rev Lett* **97**:230602.
- [77] Ooppelstrup T, Bulatov VV, Donev A, Kalos MH, Gilmer GH, Sadigh B (2009) First-passage kinetic Monte Carlo method. *Phys Rev E* **80**:066701.
- [78] Donev A, Bulatov VV, Ooppelstrup T, Gilmer GH, Sadigh B, Kalos MH (2010) A first-passage kinetic Monte Carlo algorithm for complex reaction–diffusion systems. *J Comp Phys* **229**:3214–3236.
- [79] Hellander S, Lötstedt P (2011) Flexible single molecule simulation of reaction–diffusion processes. *J Comp Phys* **230**:3948–3965.
- [80] Schwarz K, Rieger H (2013) Efficient kinetic Monte Carlo method for reaction–diffusion problems with spatially varying annihilation rates. *J Comp Phys* **237**:396–410.
- [81] Smoluchowski M (1915) Über Brownsche Molekularbewegung unter Einwirkung äußerer Kräfte und den Zusammenhang mit der verallgemeinerten Diffusionsgleichung. *Ann Phys* **353**:1103–1112.
- [82] Ross JL, Ali MY, Warshaw DM (2008) Cargo Transport: Molecular Motors Navigate a Complex Cytoskeleton. *Curr Opin Cell Biol* **20**:41–47.
- [83] Hirokawa N, Noda Y, Tanaka Y, Niwa S (2009) Kinesin superfamily motor proteins and intracellular transport. *Nat Rev Mol Cell Biol* **10**:682–696.
- [84] Verhey KJ, Hammond JW (2009) Traffic control: regulation of kinesin motors. *Nat Rev Mol Cell Biol* **10**:765–777.
- [85] Tischer C, ten Wolde PR, Dogterom M (2010) Providing positional information with active transport on dynamic microtubules. *Biophys J* **99**:726–735.
- [86] Akhmanova A, Dogterom M (2011) Kinesins Lead Aging Microtubules to Catastrophe. *Cell* **147**:966–968.
- [87] Hammer III JA, Sellers JR (2012) Walking to work: roles for class V myosins as cargo transporters. *Nat Rev Mol Cell Biol* **13**:13–26.
- [88] Hancock WO (2014) Bidirectional cargo transport: moving beyond tug of war. *Nat Rev Mol Cell Biol* **15**:615–628.
- [89] Khaitlina SY (2014) Intracellular transport based on actin polymerization. *Biochemistry Mosc* **79**:917–927.
- [90] Huber F, Boire A, López MP, Koenderink GH (2015) Cytoskeletal crosstalk: when three different personalities team up. *Current Opinion in Cell Biology* **32**:39–47.
- [91] Morelli MJ, ten Wolde PR (2008) Reaction Brownian dynamics and the effect of spatial fluctuations on the gain of a push-pull network. *J Chem Phys* **129**:054112 1–11.

- [92] Bastiaens P, Caudron M, Niethammer P, Karsenti E (2006) Gradients in the self-organization of the mitotic spindle. *Trends in Cell Biol* **16**:125–134.
- [93] Niethammer P, Bastiaens P, Karsenti E (2004) Stathmin-Tubulin Interaction Gradients in Motile and Mitotic Cells. *Science* **303**:1862–1866.
- [94] Siegrist SE, Doe CQ (2007) Microtubule-induced cortical cell polarity. *Genes & Development* **2007**:483–496.
- [95] Martin SG (2009) Microtubule-dependent cell morphogenesis in the fission yeast. *Trends Cell Biol* **19**:447–454.
- [96] Lo Presti L, Martin SG (2011) Shaping fission yeast cells by rerouting actin-based transport on microtubules. *Curr Biol* **21**:2064–2069.
- [97] Howard M (2012) How to build a robust intracellular concentration gradient. *Trends in Cell Biol* **22**:311–317.
- [98] Kruse K (2012) Bacterial Organization in Space and Time. In: Comprehensive Biophysics, Oxford: Academic Press, volume 7. pp. 208–221.
- [99] Schmick M, Bastiaens PIH (2014) The interdependence of membrane shape and cellular signal processing. *Cell* **156**:1132–1138.
- [100] Recouvreux P, Sokolowski TR, Grammoustianou A, ten Wolde PR, Dogterom M (2016) Chimera proteins with affinity for membranes and microtubule tips polarize in the membrane of fission yeast cells. *Proc Natl Acad Sci USA* **113**:1811–1816.
- [101] Bähler J, Pringle JR (1998) Pom1p, a fission yeast protein kinase that provides positional information for both polarized growth and cytokinesis. *Genes & Development* **12**:1356–1370.
- [102] Moseley JB, Mayeux A, Paoletti A, Nurse P (2009) A spatial gradient coordinates cell size and mitotic entry in fission yeast. *Nature* **459**:857–860.
- [103] Martin SG, Berthelot-Grosjean M (2009) Polar gradients of the DYRK-family kinase Pom1 couple cell length with the cell cycle. *Nature* **459**:852–856.
- [104] Tostevin F (2011) Precision of Sensing Cell Length via Concentration Gradients. *Biophys J* **100**:294–303.
- [105] Vilela M, Morgan JJ, Lindahl PA (2010) Mathematical Model of a Cell Size Checkpoint. *PLoS Comput Biol* **6**:e1001036.
- [106] Hachet O, Berthelot-Grosjean M, Kokkoris K, Vincenzetti V, Moosbrugger J, Martin SG (2011) A Phosphorylation Cycle Shapes Gradients of the DYRK Family Kinase Pom1 at the Plasma Membrane. *Cell* **145**:1116–1128.
- [107] Lochhead PA, Sibbet G, Morrice N, Cleghon V (2005) Activation-Loop Autophosphorylation Is Mediated by a Novel Transitional Intermediate Form of DYRKs. *Cell* **121**:925–936.
- [108] Lochhead PA (2009) Protein Kinase Activation Loop Autophosphorylation in Cis: Overcoming a Catch-22 Situation. *Sci Signal* **2**:pe4.
- [109] Saul VV, de la Vega L, Milanovic M, Krüger M, Braun T, Fritz-Wolf K, Becker K, Schmitz ML (2013) HIPK2 kinase activity depends on cis-autophosphorylation of its activation loop. *J Mol Cell Biol* **5**:27–38.
- [110] Hersch M, Hachet O, Dalessi S, Ullal P, Bhatia P, Bergmann S, Martin SG (2015) Pom1 gradient buffering through intermolecular auto-phosphorylation. *Mol Syst Biol* **11**:818.

- [111] Dematté L (2010) Parallel Particle-Based Reaction Diffusion: A GPU Implementation. In: 2010 Ninth International Workshop on Parallel and Distributed Methods in Verification, and Second International Workshop on High Performance Computational Systems Biology. pp. 67–77.
- [112] Gladkov DV, Alberts S, D’Souza RM, Andrews SS (2011) Accelerating the Smoldyn spatial stochastic biochemical reaction network simulator using GPUs. *Proceedings of the 19th High Performance Computing Symposia* :151–158.
- [113] Dematté L (2012) Smoldyn on graphics processing units: massively parallel Brownian dynamics simulations. *IEEE/ACM Trans Comput Biol Bioinform* **9**:655–667.
- [114] Erban R, Flegg MB, Papoian GA (2013) Multiscale Stochastic Reaction–Diffusion Modeling: Application to Actin Dynamics in Filopodia. *Bull Math Biol* **76**:799–818.
- [115] Franz B, Flegg M, Chapman S, Erban R (2013) Multiscale Reaction-Diffusion Algorithms: PDE-Assisted Brownian Dynamics. *SIAM J Appl Math* **73**:1224–1247.
- [116] Flegg M, Chapman S, Zheng L, Erban R (2014) Analysis of the Two-Regime Method on Square Meshes. *SIAM J Sci Comput* **36**:B561–B588.
- [117] Flegg MB, Hellander S, Erban R (2015) Convergence of methods for coupling of microscopic and mesoscopic reaction–diffusion simulations. *J Comput Phys* **289**:1–17.
- [118] Robinson M, Flegg M, Erban R (2014) Adaptive two-regime method: Application to front propagation. *J Chem Phys* **140**:124109.
- [119] Robinson M, Andrews SS, Erban R (2015) Multiscale reaction-diffusion simulations with Smoldyn. *Bioinformatics* :btv149.
- [120] Vijaykumar A, Bolhuis PG, ten Wolde PR (2015) Combining molecular dynamics with mesoscopic Green’s function reaction dynamics simulations. *J Chem Phys* **143**:214102.
- [121] Vijaykumar A, Ouldridge TE, ten Wolde PR, Bolhuis PG (2017) Multiscale simulations of anisotropic particles combining molecular dynamics and Green’s function reaction dynamics. *J Chem Phys* **146**:114106.
- [122] Kaizu K, Takahashi K (2017) In preparation, personal communication.
- [123] Paijmans J (2012) The fundamental lower bound of the noise in transcriptional regulation. Master’s thesis, Univeristy of Amsterdam.
- [124] Smoluchowski M (1916) Drei Vorträge über Diffusion, Brown’sche Molekularbewegung und Koagulation von Kolloidteilchen. *Physik Z* **17**:557–551 and 587–599.
- [125] Carslaw HS, Jaeger JC (1959) Conduction of Heat in Solids. Oxford University Press, 2nd edition.
- [126] Lamm G, Schulten K (1983) Extended Brownian dynamics. II. Reactive, non-linear diffusion. *J Chem Phys* **78**:2713–2734.
- [127] Watson GN (1962) A treatise on the theory of Bessel functions. Cambridge University Press, 2nd edition.
- [128] Bossen L (2010) Integrating membranes into the enhanced Green’s Function Reaction Dynamics algorithm. Master’s thesis, University of Amsterdam.
- [129] Galjart N, Perez F (2003) A plus-end raft to control microtubule dynamics and function. *Curr Opin Cell Biol* **15**:48–53.

- [130] Sawin KE (**2000**) Microtubule dynamics: The view from the tip. *Curr Biol* **10**:R860–R862.
- [131] Halford SE, Marko JF (**2004**) How do site-specific DNA-binding proteins find their targets? *Nucleic Acids Res* **32**:3040–3052.
- [132] Shimamoto N (**1999**) One-dimensional Diffusion of Proteins along DNA. *J Biol Chem* **274**:15293–15296.
- [133] von Hippel PH, Berg OG (**1989**) Facilitated target location in biological systems. *J Biol Chem* **264**:675–678.
- [134] Kivenson A, Hagan MF (**2012**). Diffusion-limited rates on low-dimensional manifolds with extreme aspect ratios. arXiv:1208.5156 [q-bio.BM].
- [135] Saunders TE, Pan KZ, Angel A, Guan Y, Shah JV, Howard M, Chang F (**2012**) Noise reduction in the intracellular pom1p gradient by a dynamic clustering mechanism. *Dev Cell* **22**:558–572.

SUPPLEMENTARY INFORMATION

S1 Algorithmic details

S1.1 The eGFRD algorithm

The box entitled “Algorithm 1” shows a pseudo-code listing of the basic eGFRD algorithm. Note that for brevity and clarity details of event (time) sampling and domain construction have been omitted here. A similar pseudo-code listing for the domain and shell construction procedure used in our new eGFRD implementation is found in box “Algorithm 2” in Section S6.

S1.2 Sampling of next-event information

In the main text, section 2.1, we introduce the *Single* and *Pair* domains featured throughout eGFRD. Here we describe in more detail how next-event times and types can be sampled from the analytical probability distributions linked to the domains.

S1.2.1 Single domains

Two types of next-events can happen within a *Single* domain \mathcal{D}_1 : either the particle undergoes a monomolecular reaction, which can mean decay or species change, or it hits the boundary of the domain by diffusion. In eGFRD, the former is called a *Single Reaction*, the latter a *Single Escape*.

If we assume that monomolecular reactions are a Poissonian process independent of particle diffusion, the propensity function for the next reaction is simply an exponential distribution

$$q_m(t) = k_m e^{-k_m t} \quad (\text{S-1})$$

where k_m is the rate of the specific monomolecular reaction. A tentative next-event time τ_m for a monomolecular reaction then can be sampled via the inversion method as

$$\tau_m = -k_m \ln(\mathcal{R}_m) \quad (\text{S-2})$$

where $\mathcal{R}_m \in [0, 1]$ is a uniformly distributed random number.

Let $p(\mathbf{r}, t|\mathbf{r}_0)$ be the density function for the probability that a diffusing particle will be located at \mathbf{r} at time t given that it started at \mathbf{r}_0 at time $t_0 = 0$. Within an unbounded volume, the evolution of $p(\mathbf{r}, t|\mathbf{r}_0)$ is well-described by the diffusion equation

$$\partial_t p(\mathbf{r}, t|\mathbf{r}_0) = D \nabla^2 p(\mathbf{r}, t|\mathbf{r}_0) + \delta(\mathbf{r} - \mathbf{r}_0) \delta(t - t_0) \quad . \quad (\text{S-3})$$

Note that due to the delta-peak inhomogeneity that represents the initial condition, the solution $p(\mathbf{r}, t|\mathbf{r}_0)$ technically is a Green’s function.

To sample a first-passage time for the particle to reach the outer shell $\partial\mathcal{D}_1$ of a domain \mathcal{D}_1 constructed around \mathbf{r}_0 , additionally an absorbing boundary condition may be imposed as follows:

$$p(\mathbf{r}, t|\mathbf{r}_0) = 0 \quad \text{for } \mathbf{r} \in \partial\mathcal{D}_1 \quad (\text{S-4})$$

In the simplest case, for a spherical domain with radius R :

$$p(|\mathbf{r} - \mathbf{r}_0| = R, t|\mathbf{r}_0) = 0 \quad (\text{S-5})$$

For more complicated domain geometries, e.g. cylinders, the problem has to be transformed into a coordinate system that captures specific symmetries, and boundary conditions have to be imposed for each coordinate separately.

Algorithm 1 Basic outline of the eGFRD algorithm. Symbols \mathcal{D}_x denote domains, τ_x next-event times. The scheduler S is the list of all next-event times in the system, ordered by increasing time. List U collects all particles that have been updated at a given time τ_x and require construction of a new domain. t_{sim} is the time that passed since simulation start.

Initialize:

$t_{\text{sim}} \leftarrow 0$, scheduler $S \leftarrow \{\}$

for all particles p_i **do**

if not p_i already in domain **then**

$\mathcal{D}_j \leftarrow$ create domain for p_i

$\tau_j \leftarrow$ draw next-event time for \mathcal{D}_j

 insert τ_j into S ordered by increasing time

end if

end for

Main loop:

while $S \neq \{\}$ **and** $t_{\text{sim}} < t_{\text{end}}$ **do**

$t_{\text{sim}} \leftarrow \tau_n =$ topmost element in S

 remove τ_n from S

 propagate \mathcal{D}_n to τ_n and remove \mathcal{D}_n

 reset particle update list: $U \leftarrow \{\}$

$U \leftarrow U \cup \{p_{n_i}\}$ **for all** particles $p_{n_i} \in \mathcal{D}_n$

while $U \neq \{\}$ **do**

$p_u \leftarrow$ next particle in U

for all domains \mathcal{D}_{u_j} close to p_u **do**

 burst: propagate \mathcal{D}_{u_j} to τ_n and remove \mathcal{D}_{u_j}

 remove τ_{u_j} from S

$U \leftarrow U \cup \{p_{u_{jk}}\}$ **for all** particles $p_{u_{jk}} \in \mathcal{D}_{u_j}$

end for

end while

for all $p_u \in U$ **do**

if not p_u already in domain **then**

$\mathcal{D}_u \leftarrow$ create domain for p_u

$\tau_u \leftarrow$ draw next-event time for \mathcal{D}_u

 insert τ_u into S

end if

end for

end while

Given that the Green's function $p(\mathbf{r}, t|\mathbf{r}_0)$ for the above boundary-value problem can be found, integration of $p(\mathbf{r}, t|\mathbf{r}_0)$ over the whole domain yields its survival probability $S(t)$, i.e. the probability for the particle to still remain within \mathcal{D}_1 at time t . Note that $S(t_0) = 1$. The survival probability is linked to the propensity function $q(t)$, which is the probability for hitting $\partial\mathcal{D}_1$ within the time interval $[t, t + dt]$, via:

$$q(t) = -\partial_t S(t) = -\partial_t \int_{\mathcal{D}_1} p(\mathbf{r}, t|\mathbf{r}_0) d\mathbf{r} \quad (\text{S-6})$$

In other words, $1 - S(t) = \int_{t_0}^t q(t') dt' = Q(t)$ is equal to the cumulative distribution function of $q(t)$ and may be used to sample a next-event time τ_e for exiting the domain via the inversion method as follows:

$$\tau_e = Q^{-1}(\mathcal{R}_e) = S^{-1}(1 - \mathcal{R}_e) \quad (\text{S-7})$$

Here $\mathcal{R}_e \in [0, 1]$ again is a uniformly distributed random number. In general, it can be difficult to calculate S^{-1} analytically. Then τ_e can be obtained by solving the equation $S(\tau_e) - \mathcal{R}_e = 0$ with a numerical rootfinder³.

After construction of an eGFRD *Single* domain, first both τ_e and τ_m are sampled as described. Since we presuppose that diffusion and monomolecular reactions are occurring independently, the next-event time for the domain is set as:

$$\tau_{\mathcal{D}_1} = \min(\tau_e, \tau_m) \quad (\text{S-8})$$

This automatically determines the event type to be either a *Single Escape* or a *Single Reaction*. For reasons discussed in section 2.1 of the main text and in section S6, eGFRD also allows for “bursting”, i.e. a “premature” update of the domain at times $\tau_b < \tau_{\mathcal{D}_1}$. Here we make use of the fact that eGFRD is capable to generate a new particle position \mathbf{r}_ν from the Green's function for an arbitrary update time $\tau_\nu \leq \tau_{\mathcal{D}_1}$. Since in these cases the PDF $p(\mathbf{r}, \tau_\nu|\mathbf{r}_0)$ is not normalized within \mathcal{D}_1 , precisely because probability leaked out through $\partial\mathcal{D}_1$ during the time $\tau_\nu - t_0$, it is important to sample \mathbf{r}_ν from the conditional PDF $p_{S(\tau_\nu)}(\mathbf{r}, \tau_\nu|\mathbf{r}_0) \equiv \frac{1}{S(\tau_\nu)} p(\mathbf{r}, \tau_\nu|\mathbf{r}_0)$. How this is done in detail depends on the geometry of the domain. For a spherical domain with radius R the angles $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi]$ in $\mathbf{r}_\nu = (r_\nu, \theta, \phi)$ are sampled from uniform distributions over the respective intervals, employing existing symmetry. If the next-event type is a *Single Escape*, $r_\nu = R$ with certainty and no further steps are required. For the other event types, i.e. *Single Reaction* or bursting, the new radial distance r_ν is sampled from

$$r_\nu = P_{S(\tau_\nu)}^{-1}(\mathcal{R}_S) \quad (\text{S-9})$$

with a uniform random number $\mathcal{R}_S \in [0, 1]$, using the cumulative conditional PDF

$$P_{S(\tau_\nu)}(r_\nu) = \frac{1}{S(\tau_\nu)} \int_0^{r_\nu} \int_0^{2\pi} \int_0^\pi p(r, \theta, \phi, \tau_\nu|\mathbf{r}_0) r^2 \sin(\theta) d\theta d\phi dr \quad . \quad (\text{S-10})$$

If a *Single Reaction* event produces two particles from one, these are put at contact at the sampled position \mathbf{r}_ν with random spatial orientation. If it is a true decay event the particle is removed from the system together with its domain. Finally, when a *Single Reaction* induces a change the particle species from s_0 to s , the s_0 particle is removed and a new s -particle is created at \mathbf{r}_ν . In any other event the particle is simply moved to \mathbf{r}_ν .

³As a matter of course, using $1 - \mathcal{R}_e$ and \mathcal{R}_e is equivalent if both are uniform random numbers from $[0, 1]$.

S1.2.2 Pair domains

Sampling of next-event times for a *Pair* domain \mathcal{D}_2 follows the same principles as for *Singles*. However, here the two particles can react at contact, which creates an additional channel of exit from the domain and a new next-event type.

Let us denote by $p_2(\mathbf{r}_A, \mathbf{r}_B, t | \mathbf{r}_{A0}, \mathbf{r}_{B0})$ the PDF for the likelihood of finding two diffusing particles A and B , initially located at positions \mathbf{r}_{A0} and \mathbf{r}_{B0} at $t = t_0$, at positions \mathbf{r}_A and \mathbf{r}_B at a later time t . The time-evolution of p_2 is governed by the SMOLUCHOWSKI equation:

$$\partial_t p_2 = [D_A \nabla_A^2 + D_B \nabla_B^2] p_2 \quad (\text{S-11})$$

Here D_A and D_B are the diffusion constants of particles A and B . As demonstrated in sec. S3 for a more general case, this problem can be simplified by transforming coordinates \mathbf{r}_A and \mathbf{r}_B to \mathbf{r} and \mathbf{R} , where \mathbf{r} is the interparticle vector and \mathbf{R} the (weighted) center-of-mass of the particles. A separation ansatz $p_2 = p_r(\mathbf{r})p_R(\mathbf{R})$ then yields two separate, uncoupled diffusion equations for \mathbf{r} and \mathbf{R} , which are equivalent to (S-11):

$$\partial_t p_r = D_r \nabla_r^2 p_r, \quad \partial_t p_R = D_R \nabla_R^2 p_R \quad (\text{S-12})$$

The constants D_r and D_R depend only on D_A and D_B . The uncoupling allows for the calculation of two Green's function solutions $p_r(\mathbf{r}, t | \mathbf{r}_0)$ and $p_R(\mathbf{R}, t | \mathbf{R}_0)$ on two subdomains \mathcal{D}_r and \mathcal{D}_R of \mathcal{D}_2 , respectively, with boundary conditions adapted to the problem as described further below. \mathcal{D}_r and \mathcal{D}_R must be defined in a way that all possible positions constructed from sampled values of \mathbf{r} and \mathbf{R} remain within the protective domain \mathcal{D}_2 . Figure S1A shows a valid definition of the subdomains for a (projected) spherical pair domain.

The Green's function p_R for the \mathbf{R} diffusion is calculated in precisely the same way as the Green's function for the one-particle problem in *Single* domains, with an absorbing boundary condition $p_R(\mathbf{R}, t) = 0$ for $\mathbf{R} \in \partial\mathcal{D}_R$. This yields a next-event time τ_R for first-arrival of \mathbf{R} to $\partial\mathcal{D}_R$, called *Center of Mass Escape* or *CoM Escape*.

Reactions between A and B are modelled via a radiating boundary condition to p_r at the particle contact radius $\sigma = R_A + R_B$:

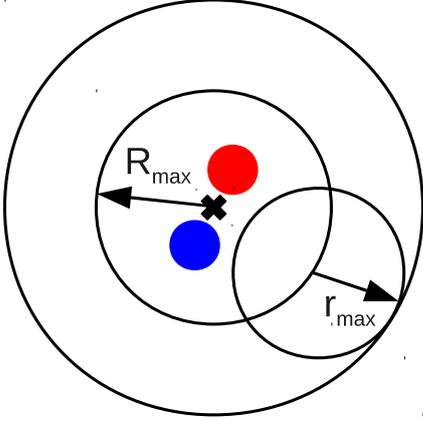
$$q_\sigma(t) \equiv \int_{\partial\mathcal{D}_r^\sigma} -D \nabla_{\mathbf{r}} p_r(\mathbf{r}, t | \mathbf{r}_0) d\mathbf{r} = k p_r(|\mathbf{r}| = \sigma, t) \quad (\text{S-13})$$

Here, k is the intrinsic particle reaction rate, which is the rate at which the particles react given that they are in contact, and $p_r(|\mathbf{r}| = \sigma, t)$ is the probability that the particles are indeed at contact at time t . The integral on the left is the total probability (out)flux through the ‘‘contact surface’’ or inner boundary of the \mathbf{r} -subdomain, which is the set of all points at which A and B are in contact: $\partial\mathcal{D}_r^\sigma = \{\mathbf{r} | |\mathbf{r}| = \sigma\}$. At the outer boundary of the \mathbf{r} -subdomain $\partial\mathcal{D}_r^a$ absorbing boundary conditions are imposed. The initial condition for this boundary value problem is set by the initial separation of the two particles, $p_r(\mathbf{r}, t = 0 | \mathbf{r}_0) = \delta(\mathbf{r} - (\mathbf{r}_{B0} - \mathbf{r}_{A0}))$. A tentative next-event time τ_r can be sampled from the survival probability $S_r(t) = \int_{\mathcal{D}_r} p_r(\mathbf{r}, t | \mathbf{r}_0) d\mathbf{r}$ in the same way as before. With this, however, it remains undetermined by which boundary the particle escaped. To specify whether the exit from \mathcal{D}_r happens through the radiating (*Pair Reaction* event) or through the absorbing boundary (*IV Escape*⁴), the probability fluxes through the boundaries at time τ_r are compared: If a uniformly distributed random number $\mathcal{R}_r \in [0, 1]$ is smaller than the fractional propensity

$$q_{\text{frac}}(\tau_r) = \frac{q_\sigma(\tau_r)}{q_\sigma(\tau_r) + q_a(\tau_r)} = \frac{q_\sigma(\tau_r)}{-\partial_t S_r(\tau_r)} \quad (\text{S-14})$$

⁴ IV = interparticle vector

A



B

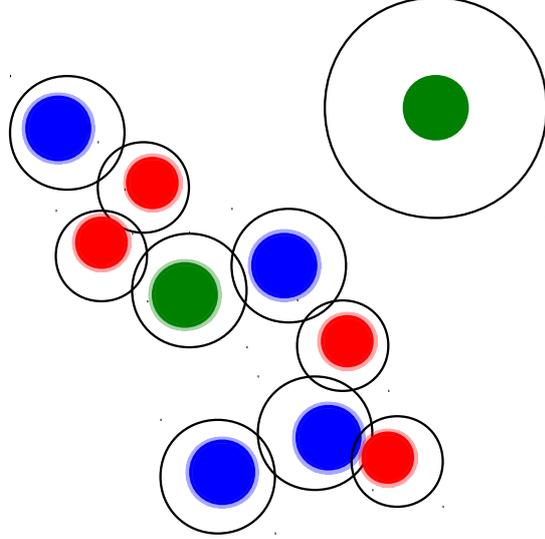


Figure S1: Pair and Multi domains in eGFRD. (A) Decomposition of a (projected) spherical *Pair* domain into subdomains for the center-of-mass vector \mathbf{R} and the interparticle vector \mathbf{r} . (B) An exemplary *Multi* domain. Here we show the situation in which particles inside the *Multi* have been already propagated by earlier updates (and thus are offset with respect to the shells), but none of them reached its outer shell yet. Thin faint-color rings around the particles indicate their reaction volume. A blue and a red particle in the lower-right part of the *Multi* overlap with their reaction volumes and will attempt a reaction. Since enough space was available, the top-right green particle formed a regular *Single* domain.

the next-event is a *Pair Reaction*; otherwise it is a *IV Escape*.

In general, the particles A and B additionally can undergo monomolecular reactions, for which next-event times τ_A and τ_B are calculated in the same manner as for *Singles*. Thus, during *Pair* domain construction, altogether four next-event times τ_r , τ_R , τ_A , τ_B with different next-event types are determined. Since the four stochastic processes again are independent of each other, the tentative next-event time for the *Pair* domain is defined as:

$$\tau_{\mathcal{D}_2} = \min(\{\tau_r, \tau_R, \tau_A, \tau_B\}) \quad (\text{S-15})$$

The precise procedure of sampling new positions for A and B at next-event time $\tau_{\mathcal{D}_2}$ depends on the type of the next event and the coordinate system in which the problem is considered. We present here the classical treatment for two interacting particles in 3D, which employs spherical coordinates, for each possible event-type respectively:

- *Pair Reaction*: Here the CoM position $\mathbf{R}_\nu(\tau_r)$ is sampled in the same manner as the new position $\mathbf{r}_\nu(\tau < \tau_{\mathcal{D}_1})$ in *Singles*. A particle with the product species is created at $\mathbf{R}_\nu(\tau_r)$.
- *IV Escape*: In this case $|\mathbf{r}_\nu| = a$ with certainty, but the escape angle θ_ν yet

remains undetermined. It is sampled from the propensity function for leaving the \mathbf{r} -subdomain at the \mathbf{r} -escape time τ_r through its outer boundary at an angle θ , given by

$$q_a(\theta) = -\frac{a \sin(\theta)}{Q} D_r \int_0^{2\pi} [\partial_r p_r(r, \theta, \phi, \tau_r | \mathbf{r}_0)]_{r=a} a d\phi .$$

Here the normalization factor is the total flux through the outer boundary $Q = \int_0^\pi q_a(\theta) d\theta$. The second angle $\phi_\nu \in [0, 2\pi]$ is drawn from a uniform distribution.

- *CoM Escape*: A new center-of-mass position $\mathbf{R}_\nu(\tau_R)$ is sampled as $\mathbf{r}_\nu(\tau_{\mathcal{D}_1})$ in the *Single*. To determine $\mathbf{r}_\nu(\tau_R)$, first a new radius r_ν is sampled from the conditional probability

$$\tilde{p}_r(r, \tau_R) \equiv \frac{r^2}{S_r(\tau_R)} \int_0^\pi \int_0^{2\pi} p_r(r, \theta, \phi, \tau_R | \mathbf{r}_0) \sin(\theta) d\theta d\phi .$$

Subsequently, a new angle θ_ν is sampled from the density $p_{r,\theta}(\theta, r_\nu, \tau_R) \equiv \frac{r_\nu \sin(\theta)}{\tilde{p}_r(r_\nu, \tau_R)} \int_0^{2\pi} p_r(r_\nu, \theta, \phi, \tau_R | \mathbf{r}_0) d\phi$ and $\phi_\nu \in [0, 2\pi]$ from the uniform distribution.

- *Monomolecular reaction* (of A)⁵: The new CoM position $\mathbf{R}_\nu(\tau_A)$ is sampled as in the case *Pair Reaction*, the new interparticle vector $\mathbf{r}_\nu(\tau_A)$ as in the case *CoM Escape*. From this we obtain $\mathbf{r}_{A,\nu}(\tau_A)$ and $\mathbf{r}_{B,\nu}(\tau_A)$. Particle B is simply moved to $\mathbf{r}_{B,\nu}(\tau_A)$, while A is treated as described for the monomolecular reactions in *Singles*.

⁵An identical procedure applies, with A and B interchanged, to the case in which B undergoes a monomolecular reaction.

S2 Brownian Dynamics fallback-system and Multi domains

The strength of eGFRD is that—thanks to the knowledge of the Green’s function—detailed sampling of diffusive trajectories inside the domains can be omitted and particles are propagated with large jumps in time and space. This, however, comes at the cost of increased computational effort per update, because drawing times and positions from Green’s functions is significantly more expensive than sampling of simple Gaussian displacements. Therefore eGFRD becomes more costly than Brownian Dynamics (BD) when particles get such crowded that the maximal size of protective domains becomes comparable to particle radii. This may be due to the presence of more than one other particle or other, static obstacles. In such situations, eGFRD seamlessly switches to a simulation mode in which particles are propagated by Brownian Dynamics within specialized domains, called *Multis*.

S2.1 Multi domains

Whenever particle distances fall below a predefined threshold and regular domain types cannot be constructed, the algorithm prompts the construction of *Multi* domains, which can contain more than two particles. An exemplary *Multi* domain is shown in Figure S1B. *Multis* are composed of intersecting spherical shells with shell radii ρ_n proportional to particle radii R_n , i.e. $\rho_n = \mu R_n$, where the “multi-shell factor” $\mu > 1$ is a simulation parameter. Each *Multi* constitutes an autonomous BD simulator isolated from its surroundings. Within their shells, particles are propagated, one at a time, by sampling displacements $\Delta\mathbf{r}$ from the free Gaussian propagator with a fixed, sufficiently small timestep Δt that ensures $|\Delta\mathbf{r}| \ll \rho_n$. Particle propagation continues until either one of the particles hits its surrounding shell or two overlapping particles react. Then, the *Multi* is broken apart and the new configuration is evaluated *de novo*, possibly resulting in *Multi* reconstruction. Particles that moved away sufficiently far from the particle crowd or obstacle at that moment reform *Single* domains and revert to propagation via Green’s functions.

When and how *Multis* are constructed is explained in more detail in section S6.

S2.2 The Reaction-Volume Method ensures that reactions in BD fulfill detailed balance

Particles that create overlaps within *Multis* are tested for reactions. Reaction events in BD mode are sampled such that detailed balance is obeyed. Let \mathbf{r}_{12} be the interparticle vector of the two interacting particles. Detailed balance demands that, for any \mathbf{r}_{12} , the probability of the unbound configuration at distance $|\mathbf{r}_{12}|$ times the transition probability to move into the bound state from \mathbf{r}_{12} equals the likelihood to be in the bound state times the probability of the inverse transition:

$$p_u(\mathbf{r}_{12})\pi_{u\rightarrow b}(\mathbf{r}_{12}) = p_b\pi_{b\rightarrow u}(\mathbf{r}_{12}) \quad (\text{S-16})$$

The occupancy ratio $p_b/p_u(\mathbf{r}_{12}) = K_{eq}$ is fixed by the equilibrium constant of the reaction and $\pi_{u\rightarrow b}(\mathbf{r}_{12})$ depends on algorithmic details of particle propagation. This leaves us with the task to prescribe a backward move in a way that $\pi_{b\rightarrow u}(\mathbf{r}_{12})$ obeys (S-16). Originally, eGFRD employed the Reaction Brownian Dynamics algorithm by MORELLI and TEN WOLDE [91]. While this scheme yields excellent results for diffusing spheres in 3D, it proved troublesome to extend it to arbitrary dimensions and non-spherical objects. In recent eGFRD we therefore implemented a conceptually similar but more versatile scheme, which we refer to as the “reaction-volume method”, or

“rvm-BD”. Its key assumption is that reactive objects, be it other particles or static structures, are surrounded by a small “reaction volume” V^* within which the precise shape of the density $p_u(\mathbf{r}_{12})$ may be ignored. Reaction attempts only occur within V^* , and at the inverse reaction the particle is placed back uniformly into V^* . The binding process is thus broken apart into a displacement and a reaction step. As in [91], it proves useful to rewrite the transition probabilities as a product of a proposal (i.e. move generation) density and (reaction) acceptance probability:

$$\begin{aligned}\pi_{u \rightarrow b}(\mathbf{r}_{12}) &= P_{u \rightarrow V^*}^{\text{gen}}(\Delta t) P_{V^* \rightarrow b}^{\text{acc}} \\ \pi_{b \rightarrow u}(\mathbf{r}_{12}) &= P_{V^* \rightarrow u}^{\text{gen}}(\Delta t) P_{b \rightarrow V^*}^{\text{acc}}\end{aligned}\tag{S-17}$$

Herein $P_{u \rightarrow V^*}^{\text{gen}}(\Delta t)$ is the probability to move diffusively into the reaction volume V^* from a distance \mathbf{r}_{12} in the unbound state within a time Δt , whereas $P_{V^* \rightarrow u}^{\text{gen}}(\Delta t)$ is the probability of the inverse move. It can be shown that these probabilities only differ by a factor V^* : $P_{V^* \rightarrow u}^{\text{gen}}(\Delta t) = V^* P_{u \rightarrow V^*}^{\text{gen}}(\Delta t)$. Together with the assumption that unbinding occurs with Poissonian statistics, i.e. $P_{b \rightarrow V^*}^{\text{acc}} = k_u \Delta t$, one finds that detailed balance is fulfilled when forward reaction attempts are accepted with a rate

$$P_{V^* \rightarrow b}^{\text{acc}} = \frac{k_b \Delta t}{V^*} .\tag{S-18}$$

In practice, V^* depends on the given situation and it is convenient to tune the magnitudes of specifically occurring reaction volumes via a global “reaction length” parameter δ . For two spherical particles with contact radius σ we have $V^* = \frac{4}{3}\pi((\sigma + \delta)^2 - \sigma^2)$. For the particle-surface interactions that we introduce later V^* is calculated similarly, taking into account the particular geometry of the contact region. This is described in more detail in [123].

The reaction length δ and the propagation time step Δt are set for each *Multi* domain individually, subject to the following two constraints:

$$D_{\text{max}} \Delta t \leq (\phi R_{\text{min}})^2\tag{S-19}$$

$$\frac{k_{\text{b,max}} \Delta t}{\delta} \leq P_{\text{max}}^{\text{acc}}\tag{S-20}$$

Here D_{max} is the maximal diffusion constant of a particle in the *Multi*, R_{min} the minimal particle radius, $k_{\text{b,max}}$ the fastest intrinsic forward rate (divided by dimension-specific contact-surface factors) and $\phi \leq 1$ a tuneable step size fraction. The first requirement limits the maximal displacement within timestep Δt to a fraction of the smallest particle size; the second ensures that the acceptance probability remains bounded by a value ≤ 1 also for fast reactions; the standard choice in eGFRD is $P_{\text{max}}^{\text{acc}} = 0.01$ and $\delta = \phi R_{\text{min}}$ with $\phi = 0.05 - 0.10$.

Note that since each *Multi* is locally constructed with a specific set of involved reaction rates and diffusion constants that does not necessarily include the fastest of them in the system, δ and Δt can be optimized in a situation-dependent manner, which is more flexible and efficient than initially setting a global time step based on the overall fastest rates.

S3 Coordinate transforms

Here we describe how the SMOLUCHOWSKI equation for two particles with independent position vectors \mathbf{r}_A and \mathbf{r}_B can be separated into two more tractable equations after transforming the original set of coordinates into new coordinates (that have the interpretation of an interparticle vector and a weighted center-of-mass or center-of-diffusion vector), and imposing a corresponding product ansatz. We will first demonstrate the calculations for a general case that takes into account a distance-dependent force between the particles; the case of constant forces acting on the two particles that we use for modeling 1d-transport in eGFRD is derived as a special case in Sec. S3.4 further below.

The SMOLUCHOWSKI equation for the probability density function $p = p(\mathbf{r}_A, \mathbf{r}_B, t | \mathbf{r}_{A0}, \mathbf{r}_{B0}, t_0)$ of two diffusing particles A and B that can interact via a force \mathbf{F} depending on their distance and move with different diffusion constants D_A and D_B is given by [81, 71]

$$\partial_t p = [D_A \nabla_A^2 + D_B \nabla_B^2 + D_A \nabla_A \cdot \varphi \mathbf{F}(\mathbf{r}) - D_B \nabla_B \cdot \varphi \mathbf{F}(\mathbf{r})] p \quad (\text{S-21})$$

where \mathbf{r} denotes the interparticle vector:

$$\mathbf{r} = \mathbf{r}_B - \mathbf{r}_A \quad (\text{S-22})$$

We define the weighted center-of-mass vector \mathbf{R} as follows:

$$\mathbf{R} = \gamma \mathbf{r}_A + \delta \mathbf{r}_B \quad (\text{S-23})$$

Equations (S-22) and (S-23) define new coordinates $\mathbf{r}(\mathbf{r}_A, \mathbf{r}_B)$ and $\mathbf{R}(\mathbf{r}_A, \mathbf{r}_B)$. Notice that this is not a coordinate transformation in the strict sense, as in general \mathbf{r} and \mathbf{R} will not be orthogonal.

Moreover, we define the operators:

$$\nabla_r \equiv \frac{\partial}{\partial \mathbf{r}} = \begin{pmatrix} \partial_{r_1} \\ \partial_{r_2} \\ \partial_{r_3} \end{pmatrix}, \quad \nabla_R \equiv \frac{\partial}{\partial \mathbf{R}} = \begin{pmatrix} \partial_{R_1} \\ \partial_{R_2} \\ \partial_{R_3} \end{pmatrix}. \quad (\text{S-24})$$

If the differential operator on the right side of (S-21) equation can be written as a sum of ∇_r^2 , ∇_r , ∇_R^2 and ∇_R , we may separate (S-21) into two independent PDEs for \mathbf{r} and \mathbf{R} by a product ansatz for p . In the following we will calculate different options for the choice of coefficients γ and δ with which the above objective is reached.

S3.1 Rewriting ∇_A and ∇_B

First we rewrite ∇_A and ∇_B in terms of ∇_r and ∇_R . Let $r_{A,i} = r_{A,i}(\mathbf{r}, \mathbf{R})$ denote the i -th component of the vector \mathbf{r}_A , and r_j, R_k components of \mathbf{r} and \mathbf{R} respectively. Then the derivative of p with respect to $r_{A,i}$ is

$$\begin{aligned} \frac{\partial p}{\partial r_{A,i}} &= \sum_j \frac{\partial p}{\partial r_j} \frac{\partial r_j}{\partial r_{A,i}} + \sum_k \frac{\partial p}{\partial R_k} \frac{\partial R_k}{\partial r_{A,i}} \\ &= \sum_j (-1) \delta_{ij} \frac{\partial p}{\partial r_j} + \sum_k \gamma \delta_{ik} \frac{\partial p}{\partial R_k} \\ &= \gamma \frac{\partial p}{\partial R_i} - \frac{\partial p}{\partial r_i} = \left(\gamma \frac{\partial}{\partial \mathbf{R}} - \frac{\partial}{\partial \mathbf{r}} \right)_i p \end{aligned} \quad (\text{S-25})$$

because r_i and R_i only depend on the component $r_{A,i}$ with the same index i . Since this holds for every i , we have:

$$\nabla_A = \gamma \nabla_R - \nabla_r \quad (\text{S-26})$$

Analogously, one obtains:

$$\nabla_B = \delta \nabla_R + \nabla_r \quad (\text{S-27})$$

From this it follows that:

$$\begin{aligned} \nabla_A^2 &= \gamma^2 \nabla_R^2 + \nabla_r^2 - 2\gamma \nabla_r \nabla_R \\ \nabla_B^2 &= \delta^2 \nabla_R^2 + \nabla_r^2 + 2\delta \nabla_r \nabla_R \end{aligned} \quad (\text{S-28})$$

Here we use $\nabla_r \nabla_R = \nabla_R \nabla_r$, assuming the 2nd derivative of p with respect to any of its variables to be a continuous function in \mathbb{R}^3 . The partial derivatives then may be interchanged by the theorem of CLAIRAUT & SCHWARZ.

Now that we have expressed ∇_A and ∇_B in terms of ∇_r and ∇_R , we can also rewrite the right side of the SMOLUCHOWSKI equation. First, for the case $\varphi = 0$, we get:

$$\begin{aligned} D_A \nabla_A^2 + D_B \nabla_B^2 &= (D_A + D_B) \nabla_r^2 \\ &\quad + (\gamma^2 D_A + \delta^2 D_B) \nabla_R^2 \\ &\quad + 2(\delta D_B - \gamma D_A) \nabla_r \nabla_R \end{aligned} \quad (\text{S-29})$$

Rewriting the force term separately yields:

$$(D_A \nabla_A - D_B \nabla_B) \cdot \varphi \mathbf{F} = [(\gamma D_A - \delta D_B) \nabla_R - (D_A + D_B) \nabla_r] \cdot \varphi \mathbf{F} \quad (\text{S-30})$$

To get rid of the mixed term containing $\nabla_r \nabla_R$ we can make any choice for γ and δ that fulfills

$$\delta = \frac{D_A}{D_B} \gamma \quad . \quad (\text{S-31})$$

Note that with this choice the force contribution (S-30) only depends on the derivative with respect to the interparticle vector (∇_r).

S3.2 Coefficient choice as in original GFRD

One possible choice for γ and δ , which is the the same as in the original version of GFRD, is the following:

$$\gamma \equiv \sqrt{\frac{D_B}{D_A}}, \quad \delta \equiv \sqrt{\frac{D_A}{D_B}} \quad (\text{S-32})$$

This yields

$$\mathbf{R} = \sqrt{\frac{D_B}{D_A}} \cdot \mathbf{r}_A + \sqrt{\frac{D_A}{D_B}} \cdot \mathbf{r}_B \quad (\text{S-33})$$

and

$$D_A \nabla_A^2 + D_B \nabla_B^2 = (D_A + D_B) (\nabla_r^2 + \nabla_R^2) \quad . \quad (\text{S-34})$$

The same prefactor also appears in the force term

$$(D_A \nabla_A - D_B \nabla_B) \cdot \varphi \mathbf{F} = -(D_A + D_B) \nabla_r \cdot \varphi \mathbf{F} \quad (\text{S-35})$$

so that equation (S-21) simplifies as follows:

$$\partial_t p = (D_A + D_B) (\nabla_r^2 + \nabla_R^2 - \nabla_r \cdot \varphi \mathbf{F}) p \quad (\text{S-36})$$

We can now separate the equation by the product ansatz

$$p(\mathbf{r}, \mathbf{R}, t | \mathbf{r}_0, \mathbf{R}_0, t_0) = p_r(\mathbf{r}, t | \mathbf{r}_0, t_0) p_R(\mathbf{R}, t | \mathbf{R}_0, t_0) \quad (\text{S-37})$$

into two equations describing two independent diffusion processes (here with the same diffusion constant $D_A + D_B$):

$$\partial_t p_r = \underbrace{(D_A + D_B)}_{D_r} (\nabla_r^2 - \nabla_r \cdot \varphi \mathbf{F}) p_r \quad (\text{S-38})$$

$$\partial_t p_R = \underbrace{(D_A + D_B)}_{D_R} \nabla_R^2 p_R \quad (\text{S-39})$$

Note that, as expected, the force contribution is present only in the equation for the interparticle vector \mathbf{r} .

S3.3 Coefficient choice as in eGFRD

The following, slightly different choice for γ and δ is used in eGFRD:

$$\gamma \equiv \frac{D_B}{D_A + D_B}, \quad \delta \equiv \frac{D_A}{D_A + D_B} \quad (\text{S-40})$$

This leads to:

$$\begin{aligned} D_A \nabla_A^2 + D_B \nabla_B^2 &= \left[\frac{D_A D_B^2 + D_B D_A^2}{(D_A + D_B)^2} \right] \nabla_R^2 + (D_A + D_B) \nabla_r^2 \\ &= \left(\frac{D_A D_B}{D_A + D_B} \right) \nabla_R^2 + (D_A + D_B) \nabla_r^2 \end{aligned} \quad (\text{S-41})$$

Everything that has been said for the previous choice of γ and δ also applies to this case, except for the fact that $\mathbf{R}(\mathbf{r}_A, \mathbf{r}_B)$ now has a different weighting as before:

$$\mathbf{R} = \frac{D_B \mathbf{r}_A + D_A \mathbf{r}_B}{D_A + D_B} \quad (\text{S-42})$$

Therefore, also the diffusion constant D_R now is different from D_r . Using the same separation ansatz as before we arrive at:

$$\begin{aligned} \partial_t p_r &= (D_A + D_B) (\nabla_r^2 - \nabla_r \cdot \varphi \mathbf{F}) p_r \\ \partial_t p_R &= \left(\frac{D_A D_B}{D_A + D_B} \right) \nabla_R^2 p_R \end{aligned} \quad (\text{S-43})$$

S3.4 Special case: Diffusion-drift equation in 1d

If there is no distance-dependent force interaction between the particles, but still constant forces that—in addition to diffusion—cause the two particles to move with average drift velocities \mathbf{v}_A and \mathbf{v}_B , the general SMOLUCHOWSKI equation, Eq. (S-21), becomes:

$$\partial_t p_2 = [D_A \nabla_A^2 + D_B \nabla_B^2 - \mathbf{v}_A \nabla_A - \mathbf{v}_B \nabla_B] p_2 \quad (\text{S-44})$$

Defining the new coordinates \mathbf{r} and \mathbf{R} and operators ∇_r and ∇_R as before [Eqs. (S-22), (S-23) and (S-24)] and once again imposing Eq. (S-31), after some intermediate steps one arrives at:

$$\begin{aligned} \partial_t p_2 &= [(D_A + D_B) \nabla_r^2 + (\gamma^2 D_A + \delta^2 D_B) \nabla_R^2 \\ &\quad - \mathbf{v}_A (\gamma \nabla_R - \nabla_r) - \mathbf{v}_B (\delta \nabla_R + \nabla_r)] p_2 \\ &= [(D_A + D_B) \nabla_r^2 + (\gamma^2 D_A + \delta^2 D_B) \nabla_R^2 \\ &\quad + (\mathbf{v}_A - \mathbf{v}_B) \nabla_r - (\gamma \mathbf{v}_A + \delta \mathbf{v}_B) \nabla_R] p_2 \end{aligned} \quad (\text{S-45})$$

Via the separation ansatz $p_2 \equiv p_r p_R$ we can rewrite the above equation in terms of two diffusion-drift equations, one for \mathbf{r} and one for \mathbf{R} , with diffusion and drift constants made up from the corresponding constants of the individual particles:

$$\begin{aligned} \partial_t p_r &= \left[\underbrace{(D_A + D_B)}_{D_r} \nabla_r^2 - \underbrace{(\mathbf{v}_B - \mathbf{v}_A)}_{\mathbf{v}_r} \nabla_r \right] p_r \\ \partial_t p_R &= \left[\underbrace{(\gamma^2 D_A + \delta^2 D_B)}_{D_R} \nabla_R^2 - \underbrace{(\gamma \mathbf{v}_A + \delta \mathbf{v}_B)}_{\mathbf{v}_R} \nabla_R \right] p_R \end{aligned} \quad (\text{S-46})$$

The interpretation of the new drift constants is straightforward: \mathbf{v}_r describes the relative velocity of the particles (as in the case without diffusive motion) while \mathbf{v}_R is an effective weighted center-of-mass drift.

Choosing “eGFRD style” coefficients γ and δ as in Sec. S3.3 we obtain:

$$D_R = \frac{D_A D_B}{D_A + D_B}, \quad \mathbf{v}_R = \frac{D_B \mathbf{v}_A + D_A \mathbf{v}_B}{D_A + D_B} \quad (\text{S-47})$$

With the definition of 1D structures and domains introduced in Sec. 2.2 of the main text, movement of the particles is restricted to a straight line. Then the vectors \mathbf{r} , \mathbf{R} , \mathbf{v}_r and \mathbf{v}_R are collinear, and we can pass from the vector equation to a scalar equation.

S3.5 Prescribing an arbitrary centre-of-mass diffusion constant

For completeness we briefly describe how to choose γ and δ in order to ensure that D_R is equal to an arbitrary prescribed diffusion constant D_C , if desired. D_C might be, for example, the diffusion constant of the product of the $A + B \rightarrow C$ reaction.

In this case γ and δ have to obey the two equations:

$$D_R = \gamma^2 D_A + \delta^2 D_B, \quad \delta = \frac{D_A}{D_B} \gamma \quad (\text{S-48})$$

Combining these we obtain:

$$D_R = \gamma^2 D_A \left[1 + \frac{D_A}{D_B} \right] \stackrel{!}{=} D_C \quad (\text{S-49})$$

Since all involved quantities are positive real numbers, it follows that:

$$\gamma = \sqrt{\frac{D_C}{D_A \left(1 + \frac{D_A}{D_B} \right)}} \quad \Rightarrow \quad \delta = \frac{D_A}{D_B} \gamma = \sqrt{\frac{D_C}{D_B \left(1 + \frac{D_B}{D_A} \right)}} \quad (\text{S-50})$$

This combination of γ and δ indeed leads to:

$$\begin{aligned} D_R = D_A \gamma^2 + D_B \delta^2 &= \frac{D_C}{1 + \frac{D_A}{D_B}} + \frac{D_C}{1 + \frac{D_B}{D_A}} \\ &= \frac{D_B D_C + D_A D_C}{D_A + D_B} = D_C \end{aligned} \quad (\text{S-51})$$

Since γ and δ are always real and positive, except for the (usually uninteresting) cases $D_A = 0$ or $D_B = 0$, one can indeed always find a coordinate transform for which D_R matches an arbitrary diffusion coefficient, while $D_r = D_A + D_B$.

S3.6 Other coordinate transforms

As we describe later in section S5, we have also devised domains for special applications, some of which feature specialized coordinate transforms; these are not described here, but introduced in the respective sections (S5.1 and S5.4).

S4 Green's function derivations and dimension-specific sampling procedures

S4.1 Green's functions for the 1D-diffusion-reaction problem with drift under different boundary conditions

After performing the coordinate transform explained in Sec. S3.4, the 1D diffusion-drift equation used for computing the Green's functions for the *Cylindrical Surface Single* and *Cylindrical Surface Pair*, introduced in sec. 2.2.4 of the main text, takes the common form

$$\partial_t p_x = [D_x \partial_x^2 - v_x \partial_x] p_x \quad (\text{PDE})$$

where x stands for either the inter-particle distance r or the center-of-mass position R . In both cases the initial condition is $p_x(x, t_0 = 0) = \delta(x - x_0)$. In the following we will drop the index and simply use $p = p(x, t|x_0, t_0)$ to denote the Green's function. Following the standard treatment in eGFRD, we model chemical reactions between particles A and B on the cylinder by imposing a radiating boundary condition to r at particle contact, while the R -equation has to be solved subject to absorbing boundary conditions. For completeness, we will also give the solutions for the half-bounded problems.

To summarize, in the following we will derive the Green's function for (PDE) on an interval $[\sigma, a]$ of length $L = a - \sigma$ or on a one-sided interval $[\sigma, \infty)$, and the following boundary conditions, respectively:

- Rad-Abs: radiating left boundary at σ , absorbing right boundary at $a > \sigma$.
- Abs-Abs: absorbing left boundary at σ , absorbing right boundary at $a > \sigma$.
- Rad-Inf: radiating left boundary at σ , no right boundary.
- Abs-Inf: absorbing left boundary at σ , no right boundary.

The Green's Functions are used to derive the resulting survival probability $S(t|x_0, t_0) = \int_a^{b \text{ or } \infty} p(x, t|x_0, t_0) dx$, the corresponding propensity function $\pi(t|x_0, t_0) = -\partial_t S(t|x_0, t_0)$ and expressions for the transient boundary fluxes.

S4.1.1 Free solution

It is easily verified that the "free", i.e. unbounded, diffusion-drift equation (PDE) with initial condition $p(x, t = 0) = \delta(x - x_0)$ is solved by

$$p_{\text{free}}(x, t|x_0) = \frac{1}{\sqrt{4\pi Dt}} e^{-\frac{1}{4Dt}[(x-x_0)-vt]^2} \quad (\text{S-52})$$

which describes a Gaussian distribution with a width that increases in time, centered around a mean value that moves with the drift velocity v .

S4.1.2 Green's function for 1D-diffusion with drift, Rad-Abs case

We start with the most general of the four cases, with the perspective of deriving other cases as special limits.

The radiation boundary condition relates the probability flux $j(x = \sigma, t)$ at the radiating boundary to the intrinsic reaction rate k via:

$$j(x = \sigma, t) = -kp(x = \sigma, t) \quad (\text{S-53})$$

Here the flux contains a contribution from diffusion and a contribution from the drift:

$$j(x, t) = -D\partial_x p(x, t) + vp(x, t) \quad (\text{S-54})$$

The correct boundary condition for the boundary at $\sigma < a$ therefore is:

$$\begin{aligned} -D\partial_x p(x, t)|_{x=\sigma} + vp(\sigma, t) &= -kp(\sigma, t) \\ \Leftrightarrow \quad \partial_x p(x, t)|_{x=\sigma} &= \frac{v+k}{D}p(\sigma, t) \end{aligned} \quad (\text{BCr})$$

The minus sign on the right side of the equation is due to the fact that at the left boundary the flux out of the system is negative with respect to the x -axis.

The absorbing boundary at $x = a$ requires:

$$p(a, t) = 0 \quad \forall t \quad (\text{BCa})$$

Dedimensionalization

Before we attempt to solve the PDE with these boundary conditions it is convenient to perform a dedimensionalization. The natural length scale is given by the length $L = a - \sigma$ of the interval $[\sigma, a]$, while $T \equiv L^2/D$ defines a corresponding natural time scale. We thus rescale our variables via

$$\begin{aligned} \xi &\equiv \frac{x - \sigma}{a - \sigma} = \frac{x - \sigma}{L}, & \Rightarrow \partial_\xi &\equiv L\partial_x \\ \tau &\equiv \frac{t}{T} = \frac{Dt}{L^2}, & \Rightarrow \partial_\tau &\equiv T\partial_t = \frac{L^2}{D}\partial_t \end{aligned} \quad (\text{S-55})$$

to obtain the following boundary value problem to solve:

$$\partial_\tau p(\xi, \tau) = \left[\partial_\xi^2 - \frac{vL}{D}\partial_\xi \right] p(\xi, \tau) \quad (\text{PDE})$$

$$\partial_\xi p(\xi, \tau)|_{\xi=0} = \frac{(v+k)L}{D}p(0, \tau) \quad (\text{BCr})$$

$$p(1, \tau) = 0 \quad (\text{BCa})$$

$$p(\xi, \tau = 0) = \frac{1}{L}\delta(\xi - \xi_0) \quad (\text{IC})$$

The last equation represents the starting condition for a particle initially located at position $L\xi_0$ (or a pair having an initial separation $L\xi_0$), where for convenience we set $t_0 = 0 = \tau_0$. Note that we have to scale the delta function by $1/L$ because the integration norm scales as $d\xi = Ldx$.

Unfortunately the linear operator $\Lambda \equiv \left[\partial_\xi^2 - \frac{vL}{D}\partial_\xi \right]$ is non-Hermitian. Therefore we can not apply straightforward techniques like eigenfunction expansion to calculate the solution. As we will see, a simple transform can resolve this issue.

Simplifying the problem with the help of an integrating factor

The difficulties imposed by the non-Hermiticity of the operator can be overcome by introducing an integrating factor $\phi(\xi) \equiv e^{\frac{v}{2D}\xi} = e^{\frac{v}{2}\xi}$ (depending explicitly on ξ)⁶. This

⁶ Λ is non-Hermitian with respect to the usual Cartesian integration norm $d\xi$. However it is Hermitian with respect to the integration norm $d\phi = \frac{v}{2}e^{\frac{v}{2}\xi}d\xi$.

technique was already used by SMOLUCHOWSKI himself [124]. Multiplying (PDE) with $1/\phi(\xi) = e^{-\frac{\nu}{2}\xi}$ and completing the square yields ($\nu \equiv \frac{vL}{D}$):

$$\begin{aligned} \partial_\tau [e^{-\frac{\nu}{2}\xi} p(\xi, \tau)] &= e^{-\frac{\nu}{2}\xi} [\partial_\xi^2 - \nu \partial_\xi] p(\xi, \tau) \\ &= \left[e^{-\frac{\nu}{2}\xi} \partial_\xi^2 - 2\frac{\nu}{2} e^{-\frac{\nu}{2}\xi} \partial_\xi + \frac{\nu^2}{4} e^{-\frac{\nu}{2}\xi} \right. \\ &\quad \left. - \frac{\nu^2}{4} e^{-\frac{\nu}{2}\xi} \right] p(\xi, \tau) \\ &= \partial_\xi^2 [e^{-\frac{\nu}{2}\xi} p(\xi, \tau)] - \frac{\nu^2}{4} e^{-\frac{\nu}{2}\xi} p(\xi, \tau) \end{aligned} \quad (\text{S-56})$$

Thus, by writing the solution with an ansatz

$$p(\xi, \tau) = \phi(\xi)\pi(\xi, \tau) = e^{\frac{\nu}{2}\xi}\pi(\xi, \tau) \quad (\text{S-57})$$

which means defining a new function

$$\pi(\xi, \tau) \equiv e^{-\frac{\nu}{2}\xi} p(\xi, \tau) \quad (\text{S-58})$$

equation (PDE) is equivalent to:

$$\partial_\tau \pi(\xi, \tau) = \partial_\xi^2 \pi(\xi, \tau) - \frac{\nu^2}{4} \pi(\xi, \tau) \quad (\text{S-59})$$

Alternatively, this can be seen by plugging the new ansatz for p into (PDE) and applying the operators accordingly.

As a next step, also the boundary conditions and the initial condition have to be transformed analogously to yield an equivalent of the whole problem. Clearly, $\pi(\xi, \tau) = 0$ whenever $p(\xi, \tau) = 0$. Thus, $\pi(\xi, \tau)$ fulfills the boundary condition at $\xi = 1$ trivially if $p(\xi, \tau)$ does so. Because of

$$\partial_\xi \pi(\xi, \tau) = [\partial_\xi e^{-\frac{\nu}{2}\xi}] p(\xi, \tau) + e^{-\frac{\nu}{2}\xi} \partial_\xi p(\xi, \tau)$$

we have

$$\begin{aligned} \partial_\xi \pi(\xi, \tau)|_{\xi=0} &= -\frac{\nu}{2} p(0, \tau) + [e^{-\frac{\nu}{2}\xi} \partial_\xi p(\xi, \tau)]_{\xi=0} \\ &= -\frac{\nu}{2} p(0, \tau) + \frac{(v+k)L}{D} p(0, \tau) \\ &= \left[\frac{\nu}{2} + \frac{kL}{D} \right] \pi(0, \tau) \end{aligned}$$

where in the last step we use $p(0, \tau) = \pi(0, \tau)$ and $\nu = \frac{vL}{D}$.

The initial condition becomes:

$$\pi(\xi, 0) = e^{-\frac{\nu}{2}\xi} p(\xi, 0) = e^{-\frac{\nu}{2}\xi} \frac{1}{L} \delta(\xi - \xi_0)$$

In the prefactor of the delta function ξ only takes values other than ξ_0 when the delta function is zero, so we can set $\xi = \xi_0$ here. This facilitates further calculations.

In summary, after multiplication with the integrating factor ϕ the initial problem for $p(\xi, \tau)$ is equivalent to the following problem for $\pi(\xi, \tau)$:

$$\partial_\tau \pi(\xi, \tau) = \left[\partial_\xi^2 - \frac{\nu^2}{4} \right] \pi(\xi, \tau) \quad (\text{PDE})$$

$$\partial_\xi \pi(\xi, \tau)|_{\xi=0} = \left[\frac{\nu}{2} + \frac{kL}{D} \right] \pi(0, \tau) \quad (\text{BCr})$$

$$\pi(1, \tau) = 0 \quad (\text{BCa})$$

$$\pi(\xi, 0) = e^{-\frac{\nu}{2}\xi_0} \frac{1}{L} \delta(\xi - \xi_0) \quad (\text{IC})$$

This result reveals that the diffusion-drift problem is mathematically equivalent to a diffusion-decay problem with a slightly modified radiating boundary condition. The strategy now is to solve (PDE) for $\pi(\xi, t)$ and reconstruct the solution $p(\xi, t)$ afterwards using (S-57).

Solving the PDE via Laplace transform

Applying the LAPLACE transform by integrating $\int_0^\infty \pi(\xi, \tau)e^{-s\tau} d\tau \equiv \hat{\pi}(\xi, s)$ on both sides of (PDE) yields:

$$\begin{aligned} s\hat{\pi}(\xi, s) - e^{-\frac{\nu}{2}\xi_0} \frac{1}{L} \delta(\xi - \xi_0) &= \left[\partial_\xi^2 - \frac{\nu^2}{4} \right] \hat{\pi}(\xi, s) \\ \Leftrightarrow \quad [\partial_\xi^2 - \kappa^2] \hat{\pi}(\xi, s) &= -\phi_0 \delta(\xi - \xi_0) \end{aligned} \quad (\text{S-60})$$

where we abbreviate $\kappa^2 \equiv \frac{\nu^2}{4} + s \geq 0$ and $\phi_0 \equiv \frac{1}{L} e^{-\frac{\nu}{2}\xi_0} > 0$. According to transformation rules, the time derivative $\partial_\tau \pi(\xi, \tau)$ converts to $\hat{\pi}(\xi, s) - \pi(\xi, \tau = 0)$ in LAPLACE space.

To solve the transformed equation we first calculate the solution of the homogenous problem. This will be used to obtain two different specific solutions on the two parts of the underlying space separated by the delta peak, i.e. $[0, \xi_0]$ and $[\xi_0, 1]$, employing the boundary conditions and a continuity/discontinuity condition at $\xi = \xi_0$. The general solution to the homogenous problem $\partial_\xi^2 \hat{\pi} = \kappa^2 \hat{\pi}(\xi, s)$ can be written as $\hat{\pi}_h(\xi, s) = \alpha \sinh(\kappa x) + \beta \cosh(\kappa x)$. We thus make an ansatz for each part of the interval $[0, 1]$ as follows:

$$\hat{\pi}(\xi, s) = \hat{\pi}_-(\xi, s) \equiv \alpha_- \sinh(\kappa \xi) + \beta_- \cosh(\kappa \xi) \quad \text{for } \xi < \xi_0 \quad (\text{S-61})$$

$$\hat{\pi}(\xi, s) = \hat{\pi}_+(\xi, s) \equiv \alpha_+ \sinh(\kappa \xi) + \beta_+ \cosh(\kappa \xi) \quad \text{for } \xi > \xi_0 \quad (\text{S-62})$$

with constant, yet arbitrary, real coefficients α_+ , β_+ and α_- , β_- . Let us first apply the absorbing boundary condition at $\xi = 1$ to (S-62):

$$\hat{\pi}_+(1) = 0 \quad \Rightarrow \quad \alpha_+ \sinh(\kappa) = -\beta_+ \cosh(\kappa) \quad (\text{S-63})$$

where we neglect the unphysical solution $\alpha_+ = 0$, $\beta_+ = 0$.

Applying the transformed radiating boundary condition at $\xi = 0$ to (S-61) yields:

$$\begin{aligned} \partial_\xi \hat{\pi}_-(\xi, s) \Big|_{\xi=0} &= \underbrace{\left[\frac{\nu}{2} + \frac{\kappa L}{D} \right]}_{\Omega} \hat{\pi}_-(0, s) \\ \Leftrightarrow [\alpha_- \kappa \cosh(\kappa \xi) + \beta_- \kappa \sinh(\kappa \xi)]_{\xi=0} &= \Omega [\alpha_- \sinh(\kappa \xi) + \beta_- \cosh(\kappa \xi)]_{\xi=0} \\ \Leftrightarrow \kappa \alpha_- &= \Omega \beta_- \end{aligned} \quad (\text{S-64})$$

Reinsertion into (S-61) and (S-62) leads to:

$$\hat{\pi}_-(\xi, s) = \alpha_- \left(\sinh(\kappa \xi) + \frac{\kappa}{\Omega} \cosh(\kappa \xi) \right) \quad (\text{S-65})$$

$$\hat{\pi}_+(\xi, s) = \alpha_+ (\sinh(\kappa \xi) - \tanh(\kappa) \cosh(\kappa \xi)) \quad (\text{S-66})$$

In order to determine coefficients α_+ and α_- we, firstly, impose continuity of $\hat{\pi}(\xi, s)$ at $\xi = \xi_0$, i.e. $\hat{\pi}_-(\xi_0, s) = \hat{\pi}_+(\xi_0, s)$ for all s . Secondly, by integrating equation (S-60)

over $[\xi_0 - \epsilon, \xi_0 + \epsilon]$ and taking the limit $\epsilon \rightarrow 0$, we obtain the following discontinuity condition for the left- and right-hand derivative $\partial_\xi \hat{\pi}(\xi, s)|_{\xi=\xi_0}$:

$$\begin{aligned} \int_{\xi_0-\epsilon}^{\xi_0+\epsilon} [\partial_\xi^2 \hat{\pi}(\xi, s) - \kappa^2 \hat{\pi}(\xi, s)] d\xi &= - \int_{\xi_0-\epsilon}^{\xi_0+\epsilon} \phi_0 \delta(\xi - \xi_0) d\xi \quad \Leftrightarrow \\ \left[\partial_\xi \hat{\pi}(\xi, s)|_{\xi_0+\epsilon} - \partial_\xi \hat{\pi}(\xi, s)|_{\xi_0-\epsilon} \right] - \kappa^2 \left[\hat{\Pi}(\xi_0 + \epsilon, s) - \hat{\Pi}(\xi_0 - \epsilon, s) \right] &= -\phi_0 \\ \xrightarrow{\epsilon \rightarrow 0} \partial_\xi \hat{\pi}_+(\xi, s)|_{\xi_0} - \partial_\xi \hat{\pi}_-(\xi, s)|_{\xi_0} &= -\phi_0 \end{aligned} \quad (\text{S-67})$$

The term $[\hat{\Pi}(\xi_0 + \epsilon, s) - \hat{\Pi}(\xi_0 - \epsilon, s)]$ vanishes for $\epsilon \rightarrow 0$ because continuity of $\hat{\pi}(\xi, s)$ at $\xi = \xi_0$ implies continuity of the stem function $\hat{\Pi}(\xi, s) = \int \hat{\pi}(\xi, s) d\xi$ at this point.

Applying the two additional constraints to (S-65) and (S-66) determines, after some algebraic steps, the coefficients α_- and α_+ :

$$\begin{aligned} \alpha_- &= \frac{-\phi_0}{\kappa \left(\frac{\kappa}{\Omega} + \tanh(\kappa) \right)} (\sinh(\kappa \xi_0) - \tanh(\kappa) \cosh(\kappa \xi_0)) \\ \alpha_+ &= \frac{-\phi_0}{\kappa \left(\frac{\kappa}{\Omega} + \tanh(\kappa) \right)} \left(\sinh(\kappa \xi_0) + \frac{\kappa}{\Omega} \cosh(\kappa \xi_0) \right) \end{aligned} \quad (\text{S-68})$$

Hence,

$$\begin{aligned} \hat{\pi}_-(\xi, s) &= \frac{-\phi_0}{\kappa \left(\frac{\kappa}{\Omega} + \tanh(\kappa) \right)} \times \\ &\quad \left(\sinh(\kappa \xi) + \frac{\kappa}{\Omega} \cosh(\kappa \xi) \right) (\sinh(\kappa \xi_0) - \tanh(\kappa) \cosh(\kappa \xi_0)) \\ \hat{\pi}_+(\xi, s) &= \frac{-\phi_0}{\kappa \left(\frac{\kappa}{\Omega} + \tanh(\kappa) \right)} \times \\ &\quad \left(\sinh(\kappa \xi_0) + \frac{\kappa}{\Omega} \cosh(\kappa \xi_0) \right) (\sinh(\kappa \xi) - \tanh(\kappa) \cosh(\kappa \xi)) \end{aligned} \quad (\text{S-69})$$

or, after multiplying numerator and denominator by $\cosh(\kappa)$:

$$\begin{aligned} \hat{\pi}_-(\xi, s) &= \\ &\quad \frac{\phi_0}{\kappa} \frac{\left(\sinh(\kappa \xi) + \frac{\kappa}{\Omega} \cosh(\kappa \xi) \right) (\sinh(\kappa) \cosh(\kappa \xi_0) - \cosh(\kappa) \sinh(\kappa \xi_0))}{\sinh(\kappa) + \frac{\kappa}{\Omega} \cosh(\kappa)} \\ \hat{\pi}_+(\xi, s) &= \\ &\quad \frac{\phi_0}{\kappa} \frac{\left(\sinh(\kappa \xi_0) + \frac{\kappa}{\Omega} \cosh(\kappa \xi_0) \right) (\sinh(\kappa) \cosh(\kappa \xi) - \cosh(\kappa) \sinh(\kappa \xi))}{\sinh(\kappa) + \frac{\kappa}{\Omega} \cosh(\kappa)} \end{aligned} \quad (\text{S-70})$$

Here we shall not forget that $\kappa = \kappa(s) = \sqrt{s + \frac{\nu^2}{4}}$.

With this we have determined unique solutions to the diffusion-drift-reaction problem for the left ($\xi \leq \xi_0$) and right ($\xi \geq \xi_0$) part of the spatial domain in LAPLACE space. Now we can attempt the back transform to the time domain, where we will find that the solution becomes symmetric in ξ and ξ_0 again.

Inverse Laplace transform via residue formula

Having $\hat{\pi}(\xi, s)$ we can obtain the corresponding function in the time domain via the BROMWICH / FOURIER-MELLIN integral:

$$\pi(\xi, t) = \mathcal{L}^{-1} [\hat{\pi}(\xi, z)] = \frac{1}{2\pi i} \lim_{T \rightarrow \infty} \int_{\gamma-iT}^{\gamma+iT} \hat{\pi}(\xi, z) e^{zt} dz \quad (\text{S-71})$$

Herein $\hat{\pi}(\xi, z)$ is the extension of $\hat{\pi}(\xi, s)$ to the complex plane. The integration has to be performed on a line perpendicular to the real axis at the positive real value γ , which must be greater than the real part of any singularity of the complex function $\hat{\pi}(\xi, z)$. Usually this is a daunting task. It is simplified a lot if $\hat{\pi}(\xi, z)$ is a holomorphic function. In that case we can apply residue calculus to compute the line integral via a contour integral. To that purpose we close the line path from $\gamma - iT$ to $\gamma + iT$ by a half-circle in the space left to it ($\{z | \text{Re}(z) \leq \gamma\}$) to obtain contour $\gamma'(T)$. In the limit $T \rightarrow \infty$ the half-circle contribution vanishes, i.e. $\lim_{T \rightarrow \infty} \oint_{\gamma'(T)} \hat{\pi}(\xi, z) dz = \lim_{T \rightarrow \infty} \int_{\gamma - iT}^{\gamma + iT} \hat{\pi}(\xi, z) dz$. The residue formula states that the integral of $\hat{\pi}(\xi, z)$ along the contour is equal to the sum of residues at the singularities of $\hat{\pi}(\xi, z)$ enclosed by the contour times $2\pi i$. Thus, if all singularities of $\hat{\pi}(\xi, z)$ are to the left of γ , the inverse LAPLACE transform can be calculated as:

$$\pi(\xi, t) = \frac{1}{2\pi i} \lim_{T \rightarrow \infty} \oint_{\gamma'(T)} \hat{\pi}(\xi, z) e^{zt} dz = \sum_n \text{Res}_{\hat{\pi}(\xi, z) e^{zt}}(z_n) \quad (\text{S-72})$$

Since $\hat{\pi}_-(\xi, s)$ and $\hat{\pi}_+(\xi, s)$ only differ by the fact that ξ and ξ_0 are interchanged it is sufficient to carry out the inverse transform for $\hat{\pi}_+(\xi, s)$ and obtain $\pi_-(\xi, t)$ by substituting $\xi \leftrightarrow \xi_0$.

As a first step we reinsert $\kappa = \sqrt{s + \frac{\nu^2}{4}}$ and substitute $s = z - \frac{\nu^2}{4}$ in $\hat{\pi}_+(\xi, s)$ where z is a complex variable now:

$$\hat{\pi}_+(\xi, z) = \frac{\phi_0}{\sqrt{z}} \frac{(\sinh(\xi_0 \sqrt{z}) + \frac{1}{\Omega} \sqrt{z} \cosh(\xi_0 \sqrt{z}))}{\sinh(\sqrt{z}) + \frac{1}{\Omega} \sqrt{z} \cosh(\sqrt{z})} \times (\sinh(\sqrt{z}) \cosh(\xi \sqrt{z}) - \cosh(\sqrt{z}) \sinh(\xi \sqrt{z})) \quad (\text{S-73})$$

Later, instead of reinserting $z = s + \frac{\nu^2}{4}$, we will apply the standard LAPLACE-inversion rule $\mathcal{L}^{-1}[\hat{f}(s + c)] = e^{-ct} f(t)$ for $c = \text{const}$.

The fact that $\hat{\pi}_+(\xi, z)$ contains \sqrt{z} which is non-holomorphic on the negative real branch recommends testing holomorphicity of the function. By TAYLOR-expanding the $\sinh(\text{const} \cdot \sqrt{z})$ and $\cosh(\text{const} \cdot \sqrt{z})$ functions one can show that $\hat{\pi}_+(\xi, z)$ can be written as a sum over purely integer powers z^n and thus is indeed a holomorphic (even entire) function. Its complex roots z_n are found by setting the denominator to zero, which yields:

$$\tanh(\sqrt{z_n}) = -\frac{1}{\Omega} \sqrt{z_n} \quad (\text{S-74})$$

It can be shown that all z_n lie on the negative real axis and that there is no singularity at $z = 0$. Since $\hat{\pi}_+(\xi, z)$ can be written as $\hat{\pi}_+(\xi, z) = g(z)/h(z)$ with functions $g(z)$ and $h(z)$ that are holomorphic in the neighborhood of each z_n , we may calculate the residue of $\hat{\pi}_+(\xi, z)$ at z_n via the well-known formula $\text{Res}_{\hat{\pi}_+(\xi, z) e^{zt}}(z_n) = g(z_n)/h'(z_n)$. Here we find, with $h(z) = \sinh(\sqrt{z}) + \frac{1}{\Omega} \sqrt{z} \cosh(\sqrt{z})$:

$$h'(z_n) = \frac{1}{2\sqrt{z_n}} \left[\left(1 + \frac{1}{\Omega} - \frac{z_n}{\Omega^2} \right) \cosh(\sqrt{z_n}) \right] \quad (\text{S-75})$$

In the next step we additionally substitute the square roots via $\sqrt{z_n} \equiv \pm i \zeta_n$ with $\zeta_n \in \mathbb{R}^+$. Taking into account $\sinh(\pm ix) = \pm \sin(x)$ and $\cosh(\pm ix) = \cos(x)$ and cancelling multiplicative minus signs, the solution in the time domain as a sum of

residues reads:

$$\begin{aligned}
\pi_+(\xi, \xi_0, t) &= \sum_n \text{Res}_{\tilde{\pi}_+(\xi, z)} e^{zt} (z_n) \\
&= e^{-\frac{\nu^2}{4}t} \cdot 2\phi_0 \sum_n e^{z_n t} \frac{\sinh(\xi z_n^{\frac{1}{2}}) + \frac{z_n^{\frac{1}{2}}}{\Omega} \cosh(\xi z_n^{\frac{1}{2}})}{\left(1 + \frac{1}{\Omega} - \frac{z_n}{\Omega^2}\right) \cosh(z_n^{\frac{1}{2}})} \\
&\quad \times \left(\sinh(z_n^{\frac{1}{2}}) \cosh(\xi_0 z_n^{\frac{1}{2}}) - \cosh(z_n^{\frac{1}{2}}) \sinh(\xi_0 z_n^{\frac{1}{2}}) \right) \\
&= -\frac{2}{L} e^{-\frac{\nu^2}{4}t - \frac{\nu}{2}\xi_0} \cdot \sum_n e^{-\zeta_n^2 t} \frac{\sin(\xi \zeta_n) + \frac{\zeta_n}{\Omega} \cos(\xi \zeta_n)}{\left(1 + \frac{1}{\Omega} + \frac{\zeta_n^2}{\Omega^2}\right) \cos(\zeta_n)} \\
&\quad \times (\sin(\zeta_n) \cos(\xi_0 \zeta_n) - \cos(\zeta_n) \sin(\xi_0 \zeta_n))
\end{aligned} \tag{S-76}$$

where the n -summation goes over the (positive) roots of the implicit equation

$$\tanh(\sqrt{z_n}) = -\frac{1}{\Omega} \sqrt{z_n} \quad \Leftrightarrow \quad \tan(\zeta_n) = -\frac{1}{\Omega} \zeta_n \tag{S-77}$$

with $\Omega = \frac{\nu}{2} + \frac{kL}{D} = \left(\frac{\nu}{2} + k\right) \frac{L}{D}$. Using the root equation (S-77) we can further simplify

$$\begin{aligned}
&\frac{(\sin(\zeta_n) \cos(\xi_0 \zeta_n) - \cos(\zeta_n) \sin(\xi_0 \zeta_n))}{\cos(\zeta_n)} = \\
&\tan(\zeta_n) \cos(\xi_0 \zeta_n) - \sin(\xi_0 \zeta_n) = -\left(\sin(\xi_0 \zeta_n) + \frac{\zeta_n}{\Omega} \cos(\xi_0 \zeta_n) \right)
\end{aligned} \tag{S-78}$$

and realize that the denominator of the summation terms in the time domain is completely symmetric in ξ and ξ_0 . Hence, the solution in the time domain is invariant to interchanging ξ and ξ_0 . After taking into account the integrating factor via $p(\xi, \xi_0, t) = e^{\frac{\nu}{2}\xi} \pi(\xi, \xi_0, t)$ and reverting dedimensionalization we can write the final solution for both sides of the spatial domain as:

$p_{\text{RA}}(x, x_0, t) \equiv$ $p(x, x_0, t) = \frac{2}{L} e^{\frac{\nu}{2D}(x-x_0) - \frac{\nu^2}{4D}t} \sum_n e^{-\frac{\zeta_n^2 D t}{L^2}} \frac{F_n(x) F_n(x_0)}{\Omega^2 + \Omega + \zeta_n^2}$ <p>with</p> $F_n(x) \equiv \Omega \sin\left(\zeta_n \frac{x - \sigma}{L}\right) + \zeta_n \cos\left(\zeta_n \frac{x - \sigma}{L}\right)$ $\Omega = \left(\frac{\nu}{2} + k\right) \frac{L}{D} \quad \text{and}$ $\zeta_n \quad \text{positive roots of} \quad \tan(\zeta_n) = -\frac{1}{\Omega} \zeta_n$	$(S-79)$ $(S-80)$ $(S-81)$ $(S-82)$
---	--

It can be easily verified that this function fulfills the imposed boundary conditions. Also the initial condition at $t = 0$ is recovered, which can be seen by expanding the delta function into the orthogonal functions $F_n(x)$ and utilizing the straightforwardly proven orthogonality relation:

$$\int_{\sigma}^a F_n(x) F_m(x) dx = \begin{cases} \frac{L}{2} (\zeta_n^2 + \Omega^2 + \Omega) & \text{for } n = m \\ 0 & \text{for } n \neq m \end{cases} \tag{S-83}$$

In the limit $v \rightarrow 0$ the solution reproduces the well-known solution for the case without drift, which can be found in [125, 14.3II, p. 360]. To verify this, set $\zeta_n = \alpha_n L$ in (S-79) and $k_1 = 1$, $k_2 = 0$, $h_1 = \frac{k}{D}$, $h_2 = 1$ in the reference formula.

Exemplary time evolution plots of $p_{\text{RA}}(x, x_0, t)$ are shown in Figure S2 for different values of diffusion coefficient D , drift velocity v and intrinsic reaction rate k .

Green's function for 1D-diffusion with drift, Abs-Abs case

From (S-79) we can easily obtain the Green's function for 1D-diffusion with drift on a finite domain with two absorbing boundaries by taking the limit $k \rightarrow \infty$. The originally radiating boundary condition at $x = \sigma$ (BCr) then becomes

$$p(\sigma, t) = \frac{D\partial_x p(x, t)|_{x=\sigma} - vp(\sigma, t)}{k} \xrightarrow{k \rightarrow \infty} 0 \quad . \quad (\text{S-84})$$

First notice that because of $\Omega = \left(\frac{v}{2} + k\right) \frac{L}{D} \xrightarrow{k \rightarrow \infty} 0$ the root equation (S-82) in the limit $k \rightarrow \infty$ reads:

$$\begin{aligned} \tan(\zeta_n) &= -\frac{1}{\Omega} \zeta_n \xrightarrow{k \rightarrow \infty} 0 \\ \Rightarrow \zeta_n &= n\pi, \quad n \in \mathbb{N} \quad \text{for } k \rightarrow \infty \end{aligned} \quad (\text{S-85})$$

To obtain the limit of (S-79) we multiply the numerator and denominator of the summation terms by $1/k^2$. Because of

$$\frac{\Omega}{k} = \left(1 + \frac{v}{2k}\right) \frac{L}{D} \xrightarrow{k \rightarrow \infty} \frac{L}{D} \quad (\text{S-86})$$

we have

$$\frac{\Omega}{k} \sin\left(\zeta_n \frac{x - \sigma}{L}\right) + \frac{\zeta_n}{k} \cos\left(\zeta_n \frac{x - \sigma}{L}\right) \xrightarrow{k \rightarrow \infty} \frac{L}{D} \sin\left(\zeta_n \frac{x - \sigma}{L}\right) \quad (\text{S-87})$$

$$\frac{\Omega^2}{k^2} + \underbrace{\frac{\Omega}{k^2} + \frac{\zeta_n^2}{k^2}}_{\rightarrow 0} \xrightarrow{k \rightarrow \infty} \frac{L^2}{D^2} \quad (\text{S-88})$$

and therefore:

$$\begin{aligned} p_{\text{AA}}(x, x_0, t) &= p_{\text{RA}}(x, x_0, t)|_{k \rightarrow \infty} \\ &= \frac{2}{L} e^{\frac{v}{2D}(x-x_0) - \frac{v^2}{4D}t} \sum_n e^{-\left(\frac{n\pi}{L}\right)^2 Dt} \sin\left(n\pi \frac{x - \sigma}{L}\right) \sin\left(n\pi \frac{x_0 - \sigma}{L}\right) \end{aligned} \quad (\text{S-89})$$

Green's function for 1D-diffusion with drift, Rad-Inf case

For completeness, we also mention here the Green's functions for the corresponding half-bounded problems. A solution for the case with only one radiating boundary at $x = 0$ and constant drift was already published by LAMM and SCHULTEN [126] and reads:

$$\begin{aligned} p_{R\infty}(x, t|x_0, t=0) &= \frac{1}{\sqrt{4\pi Dt}} \left(e^{-\frac{(x-x_0-vt)^2}{4Dt}} + e^{-\frac{vx_0}{D}} e^{-\frac{(x+x_0-vt)^2}{4Dt}} \right) \\ &\quad - \frac{v/2 + k}{D} e^{\frac{vx_0}{D}} e^{\frac{k}{D}[(x+x_0)+(k+v)t]} \operatorname{erfc}\left(\frac{x+x_0}{\sqrt{4Dt}} + \frac{v/2 + k}{D}\sqrt{Dt}\right) \end{aligned} \quad (\text{S-90})$$

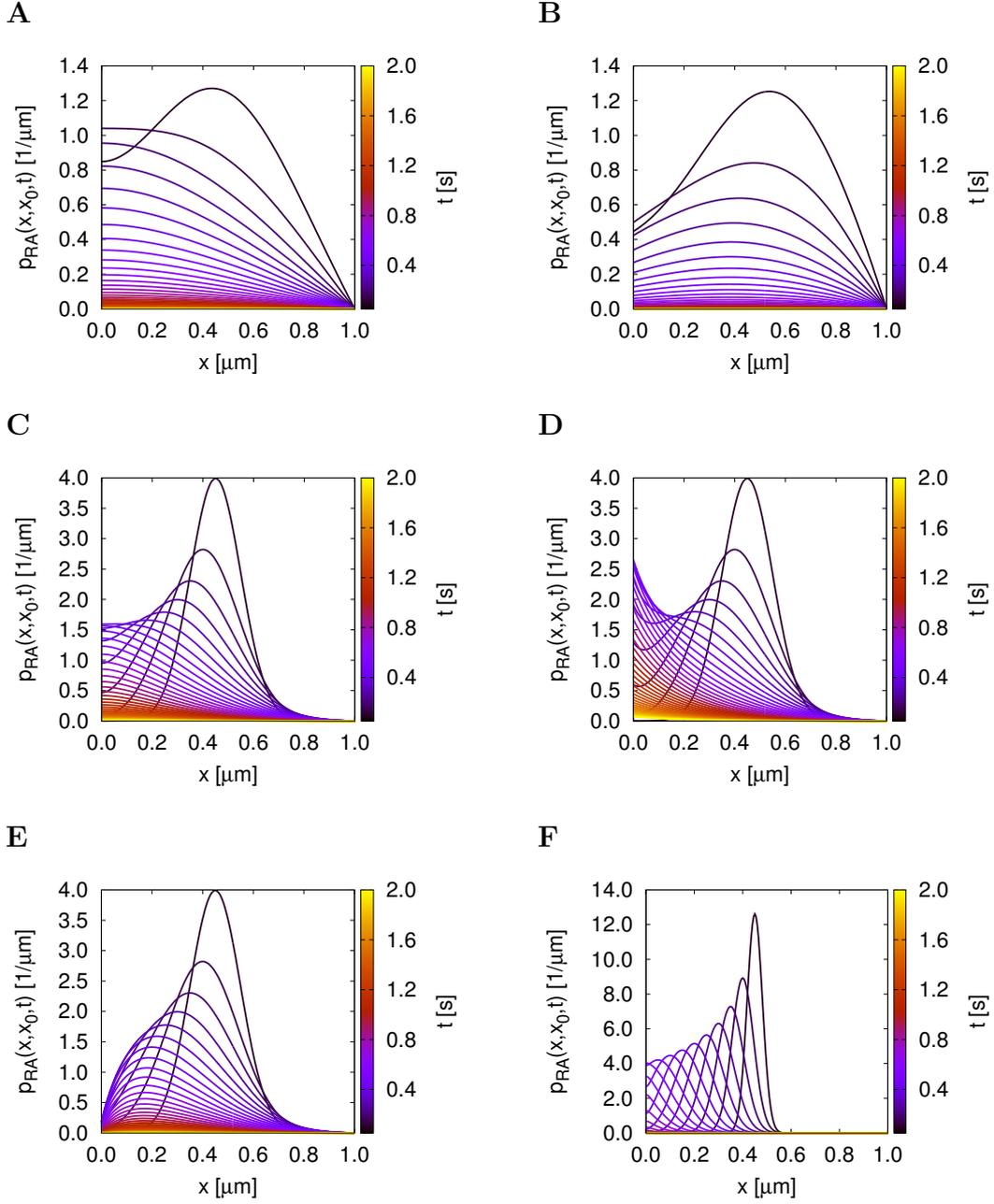


Figure S2: Green's function for the 1D diffusion-drift problem with radiating ($x = 0$) and absorbing ($x = 1$) boundary. (A) $D = 1.0$, $v = -1.0$, $k = |v|$; (B) $D = 1.0$, $v = +1.0$, $k = v$; (C) $D = 0.1$, $v = -1.0$, $k = |v|$; (D) $D = 0.1$, $v = -1.0$, $k = \frac{|v|}{2}$; (E) $D = 0.1$, $v = -1.0$, $k = 10|v|$; (F) $D = 0.01$, $v = -1.0$, $k = |v|$. Values are in $[\frac{\mu\text{m}}{\text{s}}^{(2)}]$. $x_0 = 0.5 \mu\text{m}$.

Green's function for 1D-diffusion with drift, Abs-Inf case

For the situation with only one absorbing boundary at $x = a \geq 0$ the Green's function for the 1D-diffusion-drift problem can be straightforwardly obtained via the method of images: Since any linear combination of the free solution $p_{\text{free}}(x, t|x_0)$ fulfills the diffusion-drift equation, we can easily construct a solution $p_{A\infty}(x, t|x_0)$ that will obey $p_{A\infty}(a, t|x_0) = 0 \forall t$ by subtracting from the free solution the antisymmetric solution for a particle starting from a distance $x_0 - a$ to the left of the boundary and with inverted drift $v \rightarrow -v$:

$$\begin{aligned} p_{A\infty}(x, t|x_0) &= p_{\text{free}, v_+}(x, t|x_0) - p_{\text{free}, v_-}(x, t|a - (x_0 - a)) \\ &= \frac{1}{\sqrt{4\pi Dt}} \left(e^{-\frac{1}{4D\bar{t}}[(x-x_0)-vt]^2} - e^{-\frac{1}{4D\bar{t}}[(x-2a+x_0)+vt]^2} \right) \end{aligned} \quad (\text{S-91})$$

We verified that the above solution is equivalent to the solution for this problem calculated explicitly by applying the boundary conditions in LAPLACE space and inverting via the Residue formula, following the workflow described in section S4.1.2.

Survival probabilities

The survival probability is calculated by integration of the Green's function over the whole interval on which it is defined:

$$S(t) = \int_a^b p(x, t|x_0) dx \quad (\text{S-92})$$

For the half-bounded solutions $b = \infty$. The cumulative distribution function $P(x, t|x_0) = \int_a^x p(x', t|x_0) dx'$ is needed besides $S(t)$ to sample positions at next-event times τ_ν . It is therefore convenient to first calculate $P(x, t|x_0)$ and then $S(t) = P(b, t|x_0)$ as a special case.

The Green's functions presented in this section all have the the form

$$p(x, t|x_0) = C_0 e^{\frac{v(x-x_0)}{2D}} \sum_n c_n f_n(x) \quad (\text{S-93})$$

where C_0 and c_n do not depend on x and $f_n(x)$ are either trigonometric or GAUSS functions. To calculate $P(x, t|x_0)$ the integration is most conveniently performed term-wise, i.e. by computing $\int_a^x e^{\frac{v(x'-x_0)}{2D}} f_n(x') dx'$ with the help of partial integration and reassembling the sum. Differentiation of the survival probability gives the propensity function $q(t) \equiv -\partial_t S(t)$. These are all straightforward calculations and therefore omitted here.

Boundary fluxes

With drift $v \neq 0$ the probability flux at position $x = x'$ is calculated from the Green's function $p(x, t|x_0)$ as follows:

$$q_{x'}(t) = -D\partial_x p(x, t|x_0)|_{x=x'} + vp(x', t|x_0) \quad (\text{S-94})$$

Note that for an absorbing boundary at $x' = a$ we have $p(a, t|x_0) = 0$ and the drift-dependent term vanishes.

For a radiating boundary at $x' = \sigma$ with intrinsic reaction rate k it is more convenient to calculate the flux directly from

$$q_\sigma(t) = kp(\sigma, t|x_0) \quad (\text{S-95})$$

which equals (S-94) with $x' = \sigma$ by construction of the problem.

For the Green's functions introduced in this chapter these expressions are again easily calculated and therefore not shown here. A complete collection of the survival probabilities, cumulative distribution functions and boundary fluxes for all Green's functions presented in this chapter is available as part of the eGFRD technical documentation.

S4.2 Green's function for the 2D diffusion-reaction problem

In this section we describe the derivation of the Green's function in polar coordinates $\mathbf{r} = (r, \phi)$ for a particle starting at an arbitrary position $\mathbf{r}_0 = (r_0, \phi_0)$ within an annular region bounded by a radiating inner and absorbing outer boundary. This Green's function is required for next-event sampling in the *Planar Surface Pair* domain (section 2.2.2 of the main text), but also for the *Cylindrical Surface Interaction* domain (section 2.2.3).

We assume here that the problem of two particles that interact on a plane has been transformed correctly into a diffusion problem for their center-of-mass vector \mathbf{R} and a diffusion-reaction problem for their interparticle vector $\mathbf{r} = (r, \phi)$, with a radiating boundary at particle contact, i.e. $r = |\mathbf{r}| = \sigma = R_A + R_B$, and an absorbing boundary at $r = a$. While the problem for \mathbf{R} is solved by the Green's function presented in section S4.3, the spatio-temporal evolution of \mathbf{r} is governed by the following diffusion equation in polar coordinates

$$\begin{aligned} \partial_t p_{\mathbf{r}}(r, \phi, t|r_0, \phi_0) &= D_r \nabla_{\mathbf{r}}^2 p_{\mathbf{r}}(r, \phi, t|r_0, \phi_0) \\ &= D_r \left[\partial_r^2 + \frac{1}{r} \partial_r + \frac{1}{r^2} \partial_\phi^2 \right] p_{\mathbf{r}}(r, \phi, t|r_0, \phi_0) \end{aligned} \quad (\text{PDE})$$

subject to boundary conditions

$$2\pi\sigma D_r \partial_r p_{\mathbf{r}}(r, \phi, t|r_0, \phi_0) \Big|_{r=\sigma} = kp_{\mathbf{r}}(|\mathbf{r}| = \sigma|r_0, \phi_0) \quad (\text{BCr})$$

$$p_{\mathbf{r}}(r, \phi, t|r_0, \phi_0) \Big|_{r=a} = 0 \quad (\text{BCa})$$

and initial condition

$$p_{\mathbf{r}}(r, \phi, t = 0|r_0, \phi_0) = \frac{1}{r} \delta(r - r_0) \delta(\phi - \phi_0) \quad (\text{IC})$$

where k is the intrinsic particle reaction rate.

Solution in Laplace space

The above boundary value problem is again solved most conveniently in LAPLACE space. Applying the LAPLACE transform $\hat{p}_{\mathbf{r}}(r, \phi, s|r_0, \phi_0) \equiv \int_{-\infty}^{\infty} p_{\mathbf{r}}(r, \phi, s|r_0, \phi_0) e^{-st} dt$ on both sides of the equations yields:

$$\begin{aligned} s\hat{p}_{\mathbf{r}}(r, \phi, s|r_0, \phi_0) - p_{\mathbf{r}}(r, \phi, t = 0|r_0, \phi_0) &= D_r \nabla_{\mathbf{r}}^2 \hat{p}_{\mathbf{r}}(r, \phi, s|r_0, \phi_0) \Leftrightarrow \\ \left[\partial_r^2 + \frac{1}{r} \partial_r + \frac{1}{r^2} \partial_\phi^2 - \frac{s}{D_r} \right] \hat{p}_{\mathbf{r}}(r, \phi, s|r_0, \phi_0) &= \frac{-1}{D_r r} \delta(r - r_0) \delta(\phi - \phi_0) \end{aligned} \quad (\text{PDE})$$

$$2\pi\sigma D_r \partial_r \hat{p}_{\mathbf{r}}(r, \phi, s|r_0, \phi_0) \Big|_{r=\sigma} = k\hat{p}_{\mathbf{r}}(\mathbf{r} = \sigma|r_0, \phi_0) \quad (\text{BCr})$$

$$\hat{p}_{\mathbf{r}}(r, \phi, s|r_0, \phi_0) \Big|_{r=a} = 0 \quad (\text{BCa})$$

As usual we first attempt to find a general solution to the homogenous problem corresponding to (PDE) and specialize it afterwards by applying the initial and boundary conditions. Let us set $s/D_r \equiv q^2 \geq 0$. The homogenous problem then reads:

$$\left[\partial_r^2 + \frac{1}{r} \partial_r + \frac{1}{r^2} \partial_\phi^2 - q^2 \right] \hat{p}_{\mathbf{r},h}(r, \phi, q) = 0 \quad (\text{S-96})$$

Via the separation ansatz $\hat{p}_{\mathbf{r},h}(r, \phi, q) = R(r)\Phi(\phi)$ one can show that the above PDE is equivalent to the following two differential equations coupled by a positive parameter m^2 :

$$\partial_\phi^2 \Phi(\phi) = -m^2 \Phi(\phi) \quad (\text{S-97})$$

$$[r^2 \partial_r^2 + r \partial_r - (r^2 q^2 - m^2)] R(r) \quad (\text{S-98})$$

The solution to (S-97) is readily obtained as $\Phi(\phi) = \alpha \cos(m(\phi - \phi_0))$ where we exploit that $\Phi(\phi)$ must be an even function because the operator ∂_ϕ^2 conserves the symmetry of $\delta(\phi - \phi_0)$. α is a yet undetermined real constant. With $rq \equiv \rho$ equation (S-98) is equivalent to the modified BESSEL equation:

$$[\rho^2 \partial_\rho^2 + r \partial_\rho - (\rho^2 - m^2)] R(\rho) \quad (\text{S-99})$$

which is solved by any linear combination of the modified BESSEL functions $R(\rho) = \beta I_m(\rho) + \gamma K_m(\rho)$. The solution to (S-96) thus reads

$$\begin{aligned} \hat{p}_{\mathbf{r},h}(r, \phi, q) &= (\alpha\beta) \cos(m(\phi - \phi_0)) I_m(\rho) + (\alpha\gamma) \cos(m(\phi - \phi_0)) K_m(\rho) \\ &\equiv A \cos(m(\phi - \phi_0)) I_m(qr) + B \cos(m(\phi - \phi_0)) K_m(qr) \end{aligned} \quad (\text{S-100})$$

with constants A and B .

We now can construct an ansatz for the inhomogenous problem. For further calculation it is convenient to write the ansatz as

$$\hat{p}_{\mathbf{r}} = \hat{p}_f + \hat{p}_c \quad (\text{S-101})$$

where $\hat{p}_f(\mathbf{r}, q|\mathbf{r}_0) = \frac{1}{2\pi D_r} K_0(q(\mathbf{r} \cdot \mathbf{r}_0))$ is the ‘‘free’’ solution to the unbounded 2D diffusion problem for a point particle starting at (r_0, ϕ_0) , written in LAPLACE space, and \hat{p}_c a correction resulting from the boundaries. Although \hat{p}_f fulfills the initial condition by construction, this does not automatically apply to the entire ansatz and must be separately proven later on. Since until now m^2 is an arbitrary constant, we shall construct the ansatz as a sum over all possible m . For \hat{p}_c we thus write

$$\hat{p}_c(r, \phi, q|r_0, \phi_0) = \sum_{m=-\infty}^{\infty} \cos(m(\phi - \phi_0)) [A_m I_m(qr) + B_m K_m(qr)] \quad (\text{S-102})$$

with real coefficients A_m and B_m . Now it is also convenient to expand \hat{p}_f into functions that resemble (S-100), using a formula from [125, 127, p. 365]:

$$\hat{p}_f(\mathbf{r}, q|\mathbf{r}_0) = \begin{cases} \frac{1}{2\pi D_r} \sum_{m=-\infty}^{\infty} \cos(m(\phi - \phi_0)) I_m(qr) K_m(qr_0), & r < r_0 \\ \frac{1}{2\pi D_r} \sum_{m=-\infty}^{\infty} \cos(m(\phi - \phi_0)) I_m(qr_0) K_m(qr), & r > r_0 \end{cases} \quad (\text{S-103})$$

This yields the combined ansatz:

$$\hat{p}_{\mathbf{r}}(r, \phi, q|r_0, \phi_0) = \begin{cases} \frac{1}{2\pi D_r} \sum_{m=-\infty}^{\infty} \cos(m(\phi - \phi_0)) \times \\ \quad [I_m(qr)K_m(qr_0) + A_m I_m(qr) + B_m K_m(qr)], & \text{for } r < r_0 \\ \frac{1}{2\pi D_r} \sum_{m=-\infty}^{\infty} \cos(m(\phi - \phi_0)) \times \\ \quad [I_m(qr_0)K_m(qr) + A_m I_m(qr) + B_m K_m(qr)], & \text{for } r > r_0 \end{cases} \quad (\text{S-104})$$

By applying the boundary conditions at $r = \sigma$ and $r = a$ term-wise for each m we find, after some algebraic steps

$$\begin{aligned} A_m &= K_m(qa) \frac{\mathcal{I}_m(q)K_m(qr_0) - \mathcal{K}_m(q)I_m(qr_0)}{\mathcal{K}_m(q)I_m(qa) - \mathcal{I}_m(q)K_m(qa)} \\ B_m &= \mathcal{I}_m(q) \frac{I_m(qr_0)K_m(qa) - K_m(qr_0)I_m(qa)}{\mathcal{K}_m(q)I_m(qa) - \mathcal{I}_m(q)K_m(qa)} \end{aligned} \quad (\text{S-105})$$

where we abbreviated:

$$\begin{aligned} \mathcal{I}_m(q) &= \kappa I_m(q\sigma) + qI'_m(q\sigma) \\ \mathcal{K}_m(q) &= \kappa K_m(q\sigma) + qK'_m(q\sigma) \end{aligned} \quad (\text{S-106})$$

With these coefficients the particular solution to the initial problem is completely determined in LAPLACE space.

Inverse Laplace transform

We may transform the solution back into the time domain as usual by calculating the BROMWICH integral

$$\begin{aligned} p_{\mathbf{r}}(r, \phi, t|r_0, \phi_0) &= \lim_{T \rightarrow \infty} \int_{\gamma - iT}^{\gamma + iT} \hat{p}_{\mathbf{r}}(r, \phi, s|r_0, \phi_0) e^{st} ds \\ &= \lim_{T \rightarrow \infty} \int_{\gamma - iT}^{\gamma + iT} \hat{p}_{\mathbf{r}}(r, \phi, q|r_0, \phi_0) e^{q^2 D_r t} (2D_r q) dq \end{aligned} \quad (\text{S-107})$$

where the integration as usual occurs on a line through a real constant γ located to the right of all singularities of function $\hat{p}_{\mathbf{r}}(r, \phi, q|r_0, \phi_0)$ extended to the complex plane. Assuming convergence of the sum, we can perform the integration term-wise. This somewhat lengthy calculation shall be omitted here, and we only state the final result on the next page.

$$p_{\mathbf{r}}(r, \phi, t|r_0, \phi_0) = \frac{\pi}{4} \sum_{m=-\infty}^{\infty} \cos(m(\phi - \phi_0)) \sum_{n=1}^{\infty} \frac{\zeta_{mn}^2 \mathcal{R}_m^2(\zeta_{mn}) \mathcal{J}_{mn}(r) \mathcal{J}_{mn}(r_0)}{\mathcal{R}_m^2(\zeta_{mn}) - \zeta_{mn}^2 - \kappa^2 - \frac{m^2}{\sigma^2}} e^{-\zeta_{mn}^2 D_r t} \quad (\text{S-108})$$

where ζ_{mn} are the roots of the implicit equation

$$\frac{\kappa J_m(\sigma \zeta_{mn}) - \zeta_{mn} J'_m(\sigma \zeta_{mn})}{J_m(a \zeta_{mn})} = \frac{\kappa Y_m(\sigma \zeta_{mn}) - \zeta_{mn} Y'_m(\sigma \zeta_{mn})}{Y_m(a \zeta_{mn})} \equiv \mathcal{R}_m(\zeta_{mn}) \quad (\text{S-109})$$

with $\kappa = \frac{k}{D_r}$ and

$$\mathcal{J}_{mn}(r) \equiv J_m(r \zeta_{mn}) Y_m(a \zeta_{mn}) - Y_m(r \zeta_{mn}) J_m(a \zeta_{mn}) \quad (\text{S-110})$$

As demonstrated in [128], the above function fulfills the diffusion equation (PDE), the boundary conditions (BCr) and (BCa) and the initial condition (IC).

By separating out the $m = 0$ term from the sum and unifying the summation over $m < 0$ and $m > 0$ with the help of cosine and BESSEL function (anti)symmetry relations, the Green's function (S-108) can be rewritten into a form that proves more convenient for further usage:

$$p_{\mathbf{r}}(r, \phi, t|r_0, \phi_0) = \frac{\pi}{4} \sum_{n=0}^{\infty} \frac{\zeta_{0n}^2 \mathcal{R}_0^2(\zeta_{0n}) \mathcal{J}_{0n}(r) \mathcal{J}_{0n}(r_0)}{\mathcal{R}_0^2(\zeta_{0n}) - \zeta_{0n}^2 - \kappa^2} e^{-\zeta_{0n}^2 D_r t} + \frac{\pi}{2} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \cos(m(\phi - \phi_0)) \frac{\zeta_{mn}^2 \mathcal{R}_m^2(\zeta_{mn}) \mathcal{J}_{mn}(r) \mathcal{J}_{mn}(r_0)}{\mathcal{R}_m^2(\zeta_{mn}) - \zeta_{mn}^2 - \kappa^2 - \frac{m^2}{\sigma^2}} e^{-\zeta_{mn}^2 D_r t} \quad (\text{S-111})$$

The advantage of this form of the Green's function is that here the double-sum term vanishes under the $\int_0^{2\pi} d\phi$ integral, which significantly facilitates the calculation of the survival probability and the boundary fluxes.

Survival probability

The survival probability $S(t)$ here is obtained by integrating the Green's function over the entire circular domain on which the 2D diffusion takes place:

$$S_{\mathbf{r}}(t) = \int_{\sigma}^a \int_0^{2\pi} p_{\mathbf{r}}(r, \phi, t|r_0, \phi_0) r d\phi dr \quad (\text{S-112})$$

This is done most conveniently starting from (S-111), where the part with the sum over $m > 0$ disappears under the ϕ -integral because of $\int_0^{2\pi} \cos(m(\phi - \phi_0)) d\phi = 0$. Hence, after interchanging the order of summation and integration:

$$S_{\mathbf{r}}(t) = 2\pi \cdot \frac{\pi}{4} \sum_{n=0}^{\infty} e^{-\zeta_{0n}^2 D_r t} \frac{\zeta_{0n}^2 \mathcal{R}_0^2(\zeta_{0n}) \mathcal{J}_{0n}(r_0)}{\mathcal{R}_0^2(\zeta_{0n}) - \zeta_{0n}^2 - \kappa^2} \int_{\sigma}^a r \mathcal{J}_{0n}(r) dr = \frac{\pi^2}{2} \sum_{n=0}^{\infty} e^{-\zeta_{0n}^2 D_r t} \frac{\mathcal{R}_0^2(\zeta_{0n}) \mathcal{J}_{0n}(r_0)}{\mathcal{R}_0^2(\zeta_{0n}) - \zeta_{0n}^2 - \kappa^2} \left[\frac{2}{\pi} + \sigma \mathcal{J}'_{0n}(\sigma) \right] \quad (\text{S-113})$$

$$\text{with } \mathcal{J}'_{0n}(r) = \frac{d}{dr} \mathcal{J}_{0n}(r) = \zeta_{0n} [J_1(r \zeta_{0n}) Y_0(a \zeta_{0n}) - Y_1(r \zeta_{0n}) J_0(a \zeta_{0n})]$$

To calculate the integral we have employed the well-known relations $\int r J_0(r) dr = r J_1(r)$ and $J_1'(r) = -J_0(r)$ which analogously apply to $Y_0(r)$. The $\frac{2}{\pi}$ term within the brackets originates from $\mathcal{J}'_{0n}(a) = -\frac{2}{\pi a}$ which can be shown with the help of the boundary conditions.

Boundary fluxes

We can calculate the probability fluxes through the radiating ($q_{\mathbf{r},\sigma}$) and absorbing ($q_{\mathbf{r},a}$) domain boundaries at time t by integrating the probability density gradient over the two circular contours that constitute the boundaries:

$$\begin{aligned} q_{\mathbf{r},\sigma}(t) &= \int_0^{2\pi} +D_r \partial_r p_{\mathbf{r}}(r, \phi, t | r_0, \phi_0) \Big|_{r=\sigma} r d\phi \\ q_{\mathbf{r},a}(t) &= \int_0^{2\pi} -D_r \partial_r p_{\mathbf{r}}(r, \phi, t | r_0, \phi_0) \Big|_{r=a} r d\phi \end{aligned} \quad (\text{S-114})$$

Here the signs account for opposite flux directions. In the modified form of the Green's function (S-111) again the $m \neq 0$ terms vanish under the ϕ -integral. The only r -dependent part of the $m = 0$ term is $\mathcal{J}_{0n}(r)$, the derivative of which we have calculated in (S-113). With this we arrive at:

$$\begin{aligned} q_{\mathbf{r},\sigma}(t) &= 2\pi D_r \cdot \frac{\pi}{4} \sum_{n=0}^{\infty} e^{-\zeta_{0n}^2 D_r t} \frac{\zeta_{0n}^2 \mathcal{R}_0^2(\zeta_{0n}) \mathcal{J}_{0n}(r_0)}{\mathcal{R}_0^2(\zeta_{0n}) - \zeta_{0n}^2 - \kappa^2} \sigma \mathcal{J}'_{0n}(\sigma) \\ q_{\mathbf{r},a}(t) &= 2\pi D_r \cdot \frac{\pi}{4} \sum_{n=0}^{\infty} e^{-\zeta_{0n}^2 D_r t} \frac{\zeta_{0n}^2 \mathcal{R}_0^2(\zeta_{0n}) \mathcal{J}_{0n}(r_0)}{\mathcal{R}_0^2(\zeta_{0n}) - \zeta_{0n}^2 - \kappa^2} \underbrace{(-a \mathcal{J}'_{0n}(a))}_{\frac{2}{\pi}} \end{aligned} \quad (\text{S-115})$$

S4.3 Green's function with absorbing outer boundary in polar coordinates

Here we sketch the calculation of the Green's function in polar coordinates $\mathbf{r} = (r, \phi)$ for a diffusing particle starting at radius $r = 0$ with a symmetric absorbing boundary at a radial distance $r = a$; this function is used to sample next-event times and new positions in the *Planar Surface Interaction* domain (section 2.2.1), *Planar Surface Single* and *Planar Surface Pair* domains (section 2.2.2), and the transition domains derived from the latter two (section S5.1).

Since we assume perfect radial symmetry here, the Green's function $p_s(r, t)$ does not depend on the angular coordinate ϕ , so that the corresponding boundary value problem can be written as follows:

$$\partial_t p_s(r, t) = D \left[\frac{1}{r} \partial_r (r \partial_r) \right] p_s(r, t) \quad (\text{S-116})$$

$$p_s(r, t_0 = 0) = \frac{1}{2\pi r} \delta(r) \quad (\text{S-117})$$

$$p_s(a, t) = 0 \quad (\text{S-118})$$

The solution to this problem is well-known [125, p. 368f] and, for the above initial condition, reads

$$p_s(r, t|r_0 = 0) = \frac{1}{\pi a^2} \sum_{n=1}^{\infty} e^{-\rho_n^2 D t} \frac{J_0(r \rho_n)}{J_1^2(a \rho_n)} \quad (\text{S-119})$$

where J_0 and J_1 are regular BESSEL functions and ρ_n the roots of the equation:

$$J_0(a \rho_n) = 0 \quad (\text{S-120})$$

The corresponding survival probability $S_s(t)$ and (radial) cumulative PDF $P_s(r, t)$ follow by integration of $p_s(r, t)$ over the considered circular domain:

$$S_s(t) = \int_0^a \int_0^{2\pi} p(r, t|r_0) r d\phi dr = \frac{2}{a} \sum_{n=1}^{\infty} e^{-\rho_n^2 D t} \frac{1}{\rho_n J_1(a \rho_n)} \quad (\text{S-121})$$

$$P_s(r, t) = \int_0^r \int_0^{2\pi} p(r', t|r_0) r' d\phi dr' = \frac{2}{a^2} \sum_{n=1}^{\infty} e^{-\rho_n^2 D t} \frac{r J_1(r \rho_n)}{\rho_n J_1^2(a \rho_n)} \quad (\text{S-122})$$

Here the standard formula $\int_0^r r' J_0(r') dr' = r J_1(r)$ is used.

Next-event times τ_ν are sampled from $p_s(r, t)$ in the usual way by comparing a uniform random number from $[0, 1]$ with $S_s(t)$ via the inversion method. For an arbitrary time τ , a new radius $r_\nu(\tau)$ is obtained from $\frac{1}{S_s(\tau)} P_s(r, \tau)$, whereas a new angle $\phi_\nu(\tau)$ is sampled from the uniform distribution on $[0, 2\pi]$. If $\tau = \tau_\nu$, we directly set $r_\nu(\tau_\nu) = a$.

S5 Special domains

S5.1 Plane-plane transition domains

When introducing the new domain types for membrane-interaction and -diffusion we implicitly assumed that particles interact with a single unbounded planar surface. An extension towards bounded planes does not require significant changes: domains associated to the plane must simply be constructed such that they do not reach out of it. In contrast, the implementation of particle transitions between two bordering orthogonal planes within the box-arrangement that we describe in the main text requires new domain types. We imagine that the two connected planes—in an abstract fashion—represent a continuous part of the membrane. This means that the edge does not constitute an obstacle for the diffusing particle; when it reaches the edge its movement is instantly redirected into the orthogonal direction imposed by the bordering plane. Under the assumption that this holds for each diffusive trajectory, we devised the following procedure for transitions of a particle between orthogonal planes: First we construct a spherical “*Planar Surface Transition*” domain in a way that it contains the particle on the surface of origin and an empty region on the target surface. The domain is centered around the original particle position \mathbf{r}_0 . This construction is shown in Figure S3A. As a second step a next-event time τ_ν and a new position \mathbf{r}_ν is sampled in the same way as for the *Planar Surface Single*, where the radius of the absorbing outer circle is equal to the radius of the spherical transition domain. If \mathbf{r}_ν is inside the finite plane of origin, the particle is moved to that point at τ_ν . If, in contrast, \mathbf{r}_ν lies beyond the boundaries of the original plane, the new position is deflected onto the orthogonal target plane; this is done by rotating the part of the displacement vector $\Delta\mathbf{r} = \mathbf{r}_\nu - \mathbf{r}_0$ that reaches out of the original plane about the line that marks the edge between the planes by an angle of $\pi/2$. Details of this simple geometrical transform are described in the subsequent section S5.1.1.

The above principle can be straightforwardly extended to the case of a pair of particles that reside on different neighboring planes and interact “around the edge”. Let us assume that particle A is located at position \mathbf{r}_A on plane A and particle B at \mathbf{r}_B on plane B. Here first position \mathbf{r}_B is transformed into plane A via the inverse of the deflection transform (see sec. S5.1.1). Then a next-event time and new particle positions are determined in plane A, following the procedure for the *Planar Surface Pair*. Finally, new positions that lie beyond the boundaries of plane A are transformed into plane B. Also the construction of the protective domain is slightly different as compared to the case with one particle: We encapsulate the pair constellation with a spherical “*Planar Surface Transition Pair*” domain centered around the weighted center-of-mass \mathbf{R} of the particles, as shown in Figure S3B. Note that here vector \mathbf{R} is calculated as follows: First \mathbf{r}_B is transformed into plane A, yielding \mathbf{r}'_B , and the weighted center-of-mass \mathbf{R}' is computed in plane A from \mathbf{r}'_B and \mathbf{r}_A . If \mathbf{R}' is within the bounds of plane A, we set $\mathbf{R} = \mathbf{R}'$; otherwise \mathbf{R} is obtained by deflecting \mathbf{R}' back into plane B. The latter case is shown in the example situation in Figure S3B.

Special treatment is required in the rare event that two particles end up very close to each other in the proximity of an edge between two planes. This may happen due to a single reaction, in which the products are put at contact with a random angle, but also when a *Planar Surface Transition Pair* is bursted and its two particles happen to end up close to each other. In these cases configurations are possible in which one of the particles reaches out of the plane, but application of the deflection transform would lead to particle overlap, because the transform shortens the effective distance between the two particles. Therefore particles are slightly moved apart in such situations, introducing a minor error.

S5.1.1 Deflection of particle trajectories at the edge between orthogonal planes

We will now describe the mathematical procedure that “deflects” the new position \mathbf{r}_ν of a plane-bound particle towards an adjacent, orthogonal “target” plane when \mathbf{r}_ν reaches beyond the boundaries of the original plane.

Imagine that the particle originally was located at position \mathbf{r}_0 . To transform the trajectory of the diffusing particle towards the target plane first we calculate the point \mathbf{S} at which the edge between the two planes intersects with the line $\mathbf{r}_0 + \lambda\Delta\mathbf{r}$ that links \mathbf{r}_0 and the new position \mathbf{r}_ν . Let $\hat{\mathbf{u}}_x$ and $\hat{\mathbf{u}}_z$ be the unit vectors that define the orientation of the target plane and $\hat{\mathbf{u}}_z \equiv \hat{\mathbf{u}}_x \times \hat{\mathbf{u}}_y$ the corresponding normal vector. Since \mathbf{S} lies both on the line $\mathbf{r}_0 + \lambda\Delta\mathbf{r}$ and in the target plane, it must obey

$$\mathbf{S} \cdot \hat{\mathbf{u}}_z = (\mathbf{r}_0 + \lambda_S \Delta\mathbf{r}) \cdot \hat{\mathbf{u}}_z = \mathbf{C} \cdot \hat{\mathbf{u}}_z \quad (\text{S-123})$$

where \mathbf{C} is the center point of the target plane⁷. With this we find

$$\mathbf{S} = \mathbf{r}_0 + \lambda_S \Delta\mathbf{r} \quad \text{with} \quad \lambda_S = \frac{(\mathbf{C} - \mathbf{r}_0) \cdot \hat{\mathbf{u}}_z}{\Delta\mathbf{r} \cdot \hat{\mathbf{u}}_z} \quad (\text{S-124})$$

and the protruding part of displacement vector $\Delta\mathbf{r}$:

$$\Delta\mathbf{r}' = (1 - \lambda_S)\Delta\mathbf{r} \quad (\text{S-125})$$

Instead of applying a rotation transform to $\Delta\mathbf{r}'$, here it is more convenient to construct the deflected position \mathbf{r}'_ν directly via

$$\mathbf{r}'_\nu = \mathbf{S} + \Delta r'_\parallel \hat{\mathbf{u}}_\parallel + \Delta r'_\perp \hat{\mathbf{u}}_\perp \quad (\text{S-126})$$

where $\Delta r'_\parallel = \Delta\mathbf{r}' \cdot \hat{\mathbf{u}}_\parallel$ is the component of $\Delta\mathbf{r}'$ parallel to the edge and $\Delta r'_\perp = \Delta\mathbf{r}' \cdot \hat{\mathbf{u}}_z$ its component perpendicular to the edge. $\hat{\mathbf{u}}_\parallel$ is the target plane’s unit vector parallel to the edge, whereas $\hat{\mathbf{u}}_\perp$ is the target plane’s second unit vector, which is perpendicular to both $\hat{\mathbf{u}}_\parallel$ and $\hat{\mathbf{u}}_z$. How precisely $\hat{\mathbf{u}}_\parallel$ and $\hat{\mathbf{u}}_\perp$ map onto the two unit vectors $\hat{\mathbf{u}}_x$ and $\hat{\mathbf{u}}_y$ that define the plane depends on the direction from which the particle enters the target plane. To avoid recalculation at each edge crossing, this information is stored in a neighborhood table when the box structure is constructed. It is easily proven that the deflected position is ensured to stay within the circular domain.

For the inverse transform we note that \mathbf{S} is obtained by projecting \mathbf{r}'_ν onto the original plane. With this \mathbf{r}_ν is easily constructed via:

$$\mathbf{r} = \mathbf{r}_0 + |\mathbf{r}'_\nu - \mathbf{S}| \cdot \frac{\mathbf{S} - \mathbf{r}_0}{|\mathbf{S} - \mathbf{r}_0|} \quad (\text{S-127})$$

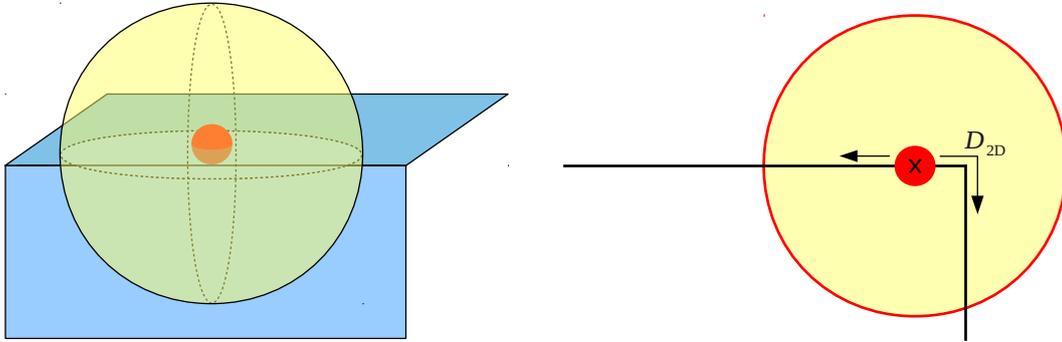
S5.2 Special domains on finite cylindrical structures

Until here we have assumed an infinite length for the cylindrical structures that particles interact with and are transported on. In many applications, however, we seek to study systems with finite 1D tracks, such as microtubules that canalize transported cargo to one of their ends preferentially, where it may unbind at a certain rate. Moreover, particles can behave differently after reaching the ends of microtubules by, for example, forming tip clusters [129, 130].

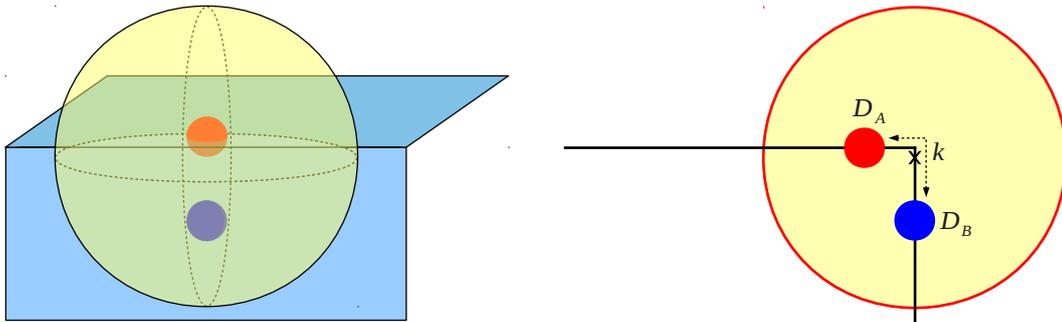
In order to include these features into eGFRD we introduce a new structure type, the “disk structure”, which is used to mark a special interaction site on a 1D cylindrical

⁷Note that instead of C , alternatively we could choose any point located in the target plane.

A



B



C

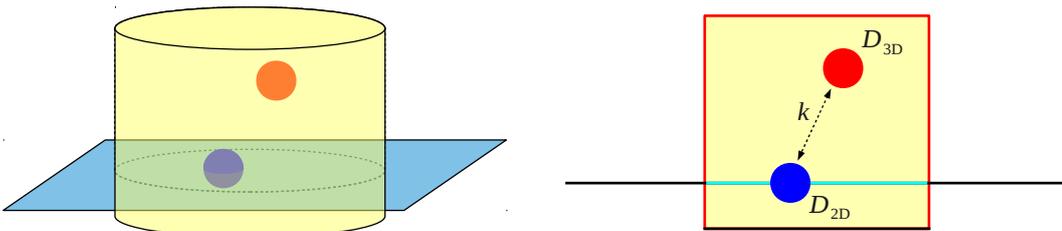


Figure S3: Special protective domains for particles on planar surfaces. (A) *Planar Surface Transition* domain; (B) *Planar Surface Transition Pair* domain; (C) *Mixed Pair 2D-3D* domain. Right panels show sections of 3D objects. Here, absorbing boundaries are highlighted by red, reflective boundaries by cyan color. The center of spherical domains is indicated by a black cross.

structure. A disk may be placed at the ends of a cylindrical structure of finite length to model reactive tip sites, or in any other place on the cylinder to model a point of interaction, e.g. a transcription-factor binding pocket on DNA. Within our framework a disk located at the end of a cylindrical structure is called a “cap”, a disk located in between the ends a “sink”. Particles on disks are immobilized. We allow particles on the cylinder to bind with a certain affinity both to disk structures and to particles already immobilized on caps. Unbinding from a cap returns the particle into the bulk, whereby the particle is moved in radial direction and placed at contact with the disk. When unbinding from a sink, the particle transfers back to the cylindrical structure, i.e. becomes mobile again. Concerning implementation, particles bound to caps or sinks are treated as individual species, which enables the definition of different parameters and reactions for cylinder-bound and disk-bound species. This may be used, for example, to introduce cap-bound species representing particle clusters that “grow” by successively absorbing particles from the cylinder via a cascade of reactions, in order to model particle clustering at filament tips.

Since the problem of a diffusing 1D particle that interacts with a disk is mathematically equivalent to the problem of two interacting particles that move in 1D, here we may re-use the 1D Green’s function with drift (S-79).

To model interactions with disks we introduce the following new domain types:

- *Cap Interaction* domain: a cylindrical domain that encloses an empty cap and a nearby cylinder-bound 1D particle. Next-event times are calculated from the 1D Green’s function with drift and Rad-Abs boundary conditions.
- *Disk Surface Single* domain: a cylindrical domain that encloses a particle bound to a disk surface. The only possible next-events are unbinding reactions which are sampled from exponential distributions.
- *Mixed Pair 1D-Cap* domain: a cylindrical domain that encloses a cap with a bound particle and a nearby cylinder-bound 1D particle. Next-event times are calculated as for the *Cylindrical Surface Pair*, with drift and diffusion coefficient of the cap-bound partner set to zero.
- *Cylindrical Surface Sink* domain: a cylindrical domain that contains a sink and a proximate cylinder-bound particle. For this special case we calculate the Green’s function explicitly in section S5.3.

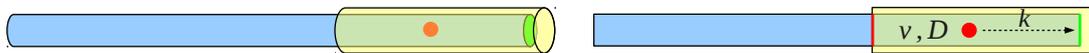
Sketches of these new domain types are shown in Figure S4.

S5.3 Cylindrical Surface Sink domain

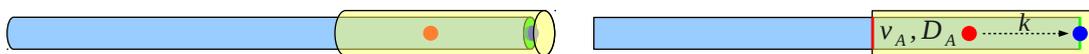
In transcription activation, transcription factors can perform a diffusive search for their binding site on the DNA [131, 132, 133]. To be able to model such and similar 1D random search processes in eGFRD, we introduced sink structures that mark point-sites at which particles can react while diffusing over the cylindrical structure that they are bound to. In order to isolate this interaction from interactions with other particles on the cylinder we further introduced new domains (*Cylindrical Surface Sink* domains) that only contain a sink and the nearby particle. Exits from these domains then can happen via two different events: either the particle hits the (absorbing) boundaries of the domain, or it binds to the sink. For the case without particle drift, we present here the Green’s function for this problem. One complication here is that the particle may diffuse over the sink without being absorbed. It is, however, possible to incorporate this feature into the mathematical derivation by imposing the correct probability flux balance at the sink position.

Assuming that the particle, starting from initial position $x = x_0$, can attach to the sink located at position x_s with an intrinsic rate k , the diffusive dynamics of the

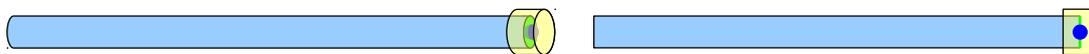
A



B



C



D

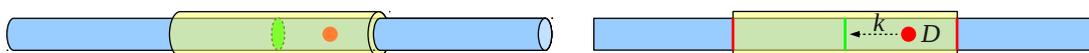


Figure S4: New protective domain types on finite 1D structures. The cylindrical surface is limited by a reactive cap at its right end (green colour). (A) *Cap Interaction* domain; (B) *Mixed Pair 1D-Cap* domain; (C) *Disk Surface Single* domain, here shown for a particle on a cap; (D) *Cylindrical Surface Sink* domain. Right panels show sections of 3D objects along the common cylinder axis. Absorbing boundaries are highlighted by red, radiating boundaries by green. Note that drift velocities (v , v_A , v_B) can be towards any cylinder end.

particle may be described via the modified diffusion equation:

$$\partial_t p(x, t|x_0) = D\nabla^2 p(x, t|x_0) - k\delta(x - x_s)p(x, t|x_0) \quad (\text{S-128})$$

with absorbing boundaries at $x = a$ and $x = b > x_s > a$

$$p(a, t|x_0) = 0, \quad p(b, t|x_0) = 0 \quad (\text{S-129})$$

and initial condition

$$p(x, t = 0|x_0) = \delta(x - x_0) \quad . \quad (\text{S-130})$$

As with the 1D-Rad-Abs Green's function calculated in section S4.1.2, this problem may be solved separately for the subintervals of $[a, b]$ separated by the delta peaks, imposing continuity of $p(x, t|x_0)$ and discontinuity of its derivative at the junction points. Here it is convenient to consider the intervals left and right to the sink and to account for the initial condition directly by an adequate ansatz for the part that contains the starting point x_0 . Continuity-discontinuity relations then only have to be imposed at x_s .

Let us denote the solution on interval $[a, x_s]$ by $p_-(x, t|x_0)$ and the solution on $[x_s, b]$ by $p_+(x, t|x_0)$. By integrating (S-128) over an ϵ -interval around the sink and taking the limit $\epsilon \rightarrow 0$ we obtain the discontinuity condition for the probability flux at $x = x_s$:

$$\partial_x p_-(x, t|x_0)|_{x=x_s} - \partial_x p_+(x, t|x_0)|_{x=x_s} = -\frac{k}{D}p_-(x_s, t|x_0) \quad (\text{S-131})$$

This equation simply states that the flux from/towards the left of the sink equals the flux towards/from the region right of it, minus the reactive flux through the sink. Moreover, continuity requires:

$$p_-(x_s, t|x_0) = p_+(x_s, t|x_0) \quad (\text{S-132})$$

Once more, this problem is most conveniently solved in LAPLACE space. The homogenous version of the LAPLACE-transformed PDE reads:

$$s\hat{p}(x, s|x_0) = D\nabla^2 \hat{p}(x, s|x_0) \quad (\text{S-133})$$

Let us without loss of generality assume $x_0 \in [x_s, b]$ and that the sink is located at $x = x_s = 0$, implying $a = -|a| < 0$. Then we can make the following ansatz in LAPLACE space ($q \equiv \sqrt{\frac{s}{D}}$):

$$\begin{aligned} \hat{p}_-(x, s|x_0) &= A_- \sinh(qx) + B_- \cosh(qx), & x < 0 \\ \hat{p}_+(x, s|x_0) &= A_+ \sinh(qx) + B_+ \cosh(qx) + \underbrace{\frac{1}{2Dq} e^{-q|x-x_0|}}_{\hat{p}_{\text{free}}(x, s|x_0)}, & x > 0 \end{aligned} \quad (\text{S-134})$$

Function $\hat{p}_+(x, s|x_0)$ contains the (LAPLACE-transformed) free solution for a point particle starting from $x = x_0$ and thus fulfills the initial condition by construction.

The coefficients A_{\pm} and B_{\pm} are calculated by applying the boundary and continuity-discontinuity conditions. Subsequently, the solution can be transformed back into the time-domain via the residue formula. This procedure is precisely the same as in S4.1.2 and therefore omitted here.

The final solution reads, with $L \equiv b - a = b + |a|$,

$$p_-(x, t|x_0) = -2D \sum_{n=1}^{\infty} e^{-D\zeta_n^2 t} \sin(\zeta_n(|a| + x)) \frac{\zeta_n \sin(\zeta_n(b - x_0))}{\Xi_n} \quad (\text{S-135})$$

$$p_+(x, t|x_0) = -2D \sum_{n=1}^{\infty} e^{-D\zeta_n^2 t} \sin(\zeta_n(b - \hat{x})) \times \frac{D\zeta_n \sin(\zeta_n(|a| + \check{x})) + k \sin(\zeta_n(|a|) \sin(\zeta_n \check{x}))}{\Xi_n} \quad (\text{S-136})$$

with $\hat{x} \equiv \max(x, x_0)$, $\check{x} \equiv \min(x, x_0)$ and a common denominator:

$$\Xi_n = D [L\zeta_n \cos(\zeta_n L) + \sin(\zeta_n L)] + k [b \cos(\zeta_n b) \sin(\zeta_n |a|) + |a| \cos(\zeta_n |a|) \sin(\zeta_n b)] \quad (\text{S-137})$$

The numbers ζ_n are all real and positive and the roots of the equation

$$D\zeta_n \sin(\zeta_n L) + k \sin(\zeta_n |a|) \sin(\zeta_n b) = 0 \quad (\text{S-138})$$

which, with the help of trigonometric relations and setting $\Delta L \equiv b - |a|$, may be written in the more convenient form:

$$D\zeta_n \sin(\zeta_n L) = \frac{k}{2} [\cos(\zeta_n L) - \cos(\zeta_n \Delta L)] \quad (\text{S-139})$$

The necessity of interchanging x and x_0 in (S-136) when the sign of $(x - x_0)$ changes arises from the presence of $|x - x_0|$ in the ansatz for $p_+(x, s|x_0)$.

The survival probability for the whole domain $[a, b]$ is given by

$$S(t) = \int_a^{x_s=0} p_-(x, t|x_0) dt + \int_{x_s=0}^b p_+(x, t|x_0) dt \quad (\text{S-140})$$

and easily obtained by simple integration.

The next-event time τ_ν for the associated *Cylindrical Surface Sink* domain is sampled from $1 - S(t)$, as usual. The next-event type, i.e. whether the particle exits the domain by being absorbed at the sink or at one of the boundaries, is determined by comparing the probability fluxes through these exit channels at τ_ν . These are most conveniently calculated via:

$$\begin{aligned} q_s &= kp_-(x_s, \tau_\nu|x_0) = kp_+(x_s, \tau_\nu|x_0) \\ q_a &= -D\partial_x p_-(x, \tau_\nu|x_0)|_{x=a} \\ q_b &= -D\partial_x p_+(x, \tau_\nu|x_0)|_{x=b} \end{aligned} \quad (\text{S-141})$$

S5.4 Mixed Pair 2D-3D domain

In order to be able to efficiently simulate a binding reaction of a particle diffusing in 3d to a particle diffusing on a 2d plane in eGFRD we devised a special domain type, the *Mixed Pair 2D-3D* domain, shown in Fig. S3C; it is based on a coordinate transform that takes into account the anisotropy in the diffusion behavior of the two reactants, by essentially rescaling the interparticle vector in the direction perpendicular to the plane. Once the particle positions have been transformed into the associated coordinate system, Green's functions already derived for the more simple situations in 2D and 3D can be used to sample next-event times and new positions for the reactants; subsequently, these can be converted back to the original coordinate transform via a simple inverse transform.

In the following we first derive the generalized coordinate transform for the direct binding scenario, and then describe how it can be used to implement the *Mixed Pair 2D-3D* domain.

S5.4.1 Generalized linear coordinate transform for direct binding

In this section we present a generalization of the linear coordinate transform that maps two arbitrary particle positions \mathbf{r}_A and \mathbf{r}_B onto a (generalized) interparticle vector \mathbf{r} and weighted center-of-mass vector \mathbf{R} in a way that makes the diffusion anisotropy arising in the direct binding scenario disappear in the transformed coordinates. To this end we pursue the following approach: First, we rewrite the linear operator (Laplacian) of the diffusion equation in a generic matrix notation in order to account for anisotropic diffusion, starting from the well-known form of the SMOLUCHOWSKI equation. As a second step, we write down the linear coordinate transform in its most general form and apply it to both the linear operator and the original coordinates \mathbf{r}_A and \mathbf{r}_B in order to decouple the equation. Finally, the generic result will be used to specify a particular transform for the direct binding scenario. This involves some freedom in the choice of transformation coefficients. We therefore postulate the following criteria to constrain the result: First, the transformed Laplacian should not contain any mixed derivatives because this complicates the solution of the transformed equation. Second, the new coordinates should capture existing symmetries and, moreover, resemble the previously defined interparticle vector and center of mass as much as possible, so that we can use previously derived Green's functions.

Rewriting the Laplace operator in matrix notation

Let us recall the SMOLUCHOWSKI equation for the density of the probability $p \equiv p(\mathbf{r}_A, \mathbf{r}_B, t | \mathbf{r}_{A0}, \mathbf{r}_{B0}, t_0)$ to find two diffusing particles A and B with diffusion constants D_A and D_B at positions \mathbf{r}_A and \mathbf{r}_B , given that they started from \mathbf{r}_{A0} and \mathbf{r}_{B0} (compare to section S1.2.2):

$$\partial_t p = [D_A \nabla_A^2 + D_B \nabla_B^2] p \quad (\text{S-142})$$

Here for simplicity we neglect the force-interaction term, i.e. $F(\mathbf{r}_A - \mathbf{r}_B) = 0$.

By introducing

$$\nabla_X \equiv \begin{pmatrix} \nabla_A \\ \nabla_B \end{pmatrix} \quad (\text{S-143})$$

and its transpose $\nabla_X^T \equiv (\nabla_A, \nabla_B)$ the linear operator $\mathcal{L}^2 \equiv D_A \nabla_A^2 + D_B \nabla_B^2$ may be written in vector-matrix-notation as

$$\mathcal{L}^2 = \nabla_X^T \begin{pmatrix} D_A & \\ & D_B \end{pmatrix} \nabla_X \equiv \nabla_X^T \mathcal{D} \nabla_X \quad (\text{S-144})$$

where the sub-matrices

$$\mathcal{D}_{A(B)} \equiv \begin{pmatrix} D_{A(B)} & & \mathbb{O} \\ & D_{A(B)} & \\ \mathbb{O} & & D_{A(B)} \end{pmatrix} \quad (\text{S-145})$$

define a diffusion matrix \mathcal{D} . While here the entries of matrices \mathcal{D}_A and \mathcal{D}_B are equal along their diagonals, in general they may differ if diffusion is anisotropic. Importantly, the Laplacian \mathcal{L}^2 has no mixed derivatives only if \mathcal{D} is diagonal.

Writing the linear coordinate transform in matrix notation

A generic linear coordinate transform $\mathcal{M} : \mathbb{R}^6 \rightarrow \mathbb{R}^6$ for two arbitrary \mathbb{R}^3 -vectors \mathbf{r}_A and \mathbf{r}_B is described via:

$$\mathbf{r} \equiv a\mathbf{r}_A + b\mathbf{r}_B, \quad \mathbf{R} \equiv c\mathbf{r}_A + d\mathbf{r}_B. \quad (\text{S-146})$$

In matrix notation this reads

$$\mathbf{Y} \equiv \begin{pmatrix} \mathbf{r} \\ \mathbf{R} \end{pmatrix} = \mathcal{M} \begin{pmatrix} \mathbf{r}_A \\ \mathbf{r}_B \end{pmatrix} \equiv \mathcal{M}\mathbf{X} \quad (\text{S-147})$$

with

$$\mathcal{M} = \begin{pmatrix} a & 0 & 0 & b & 0 & 0 \\ 0 & a & 0 & 0 & b & 0 \\ 0 & 0 & a & 0 & 0 & b \\ c & 0 & 0 & d & 0 & 0 \\ 0 & c & 0 & 0 & d & 0 \\ 0 & 0 & c & 0 & 0 & d \end{pmatrix} \equiv \begin{pmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{pmatrix}. \quad (\text{S-148})$$

We may generalize this transform further by allowing the nonzero coefficients to differ from each other as long as the full rank of the matrix is preserved. In the following we therefore assume that \mathcal{M} has the form:

$$\mathcal{M} = \begin{pmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{pmatrix} = \begin{pmatrix} a_1 & 0 & 0 & b_1 & 0 & 0 \\ 0 & a_2 & 0 & 0 & b_2 & 0 \\ 0 & 0 & a_3 & 0 & 0 & b_3 \\ c_1 & 0 & 0 & d_1 & 0 & 0 \\ 0 & c_2 & 0 & 0 & d_2 & 0 \\ 0 & 0 & c_3 & 0 & 0 & d_3 \end{pmatrix} \quad (\text{S-149})$$

Transforming the Laplace operator

Let us now apply the generalized transform to the linear operator \mathcal{L}^2 . This means expressing the derivatives ∂_{X_i} of the original coordinates in terms of derivatives ∂_{Y_i} of the new, transformed coordinates. The chain rule of differentiation yields:

$$\frac{\partial}{\partial X_i} = \frac{\partial Y_j}{\partial X_i} \frac{\partial}{\partial Y_j} \equiv N_{ij} \frac{\partial}{\partial Y_j} \quad (\text{S-150})$$

The coefficients $N_{ij} = \frac{\partial Y_j}{\partial X_i}$ define a new matrix \mathcal{N} . Since the considered transform is linear these coefficients must be constants and related to the entries of the matrix \mathcal{M} via

$$\frac{\partial Y_i}{\partial X_j} = (\mathcal{M})_{ij} = N_{ji}. \quad (\text{S-151})$$

Since for a linear transform the Jacobian and the matrix of the transform are identical, we have:

$$\begin{aligned}\mathcal{N} &= \mathcal{M}^T \\ \nabla_X &= \mathcal{M}^T \nabla_Y\end{aligned}\tag{S-152}$$

With this we may rewrite the linear operator as follows⁸:

$$\begin{aligned}\mathcal{L}^2 &= \nabla_X^T \begin{pmatrix} \mathcal{D}_A & \\ & \mathcal{D}_B \end{pmatrix} \nabla_X \\ &= \nabla_Y^T \mathcal{M} \begin{pmatrix} \mathcal{D}_A & \\ & \mathcal{D}_B \end{pmatrix} \mathcal{M}^T \nabla_Y \equiv \nabla_Y^T \mathcal{D}' \nabla_Y \equiv \mathfrak{L}^2\end{aligned}\tag{S-153}$$

Here \mathcal{D}' is the transformed diffusion matrix in the new coordinates. Recall that \mathfrak{L}^2 will not contain mixed derivatives after the coordinate transform only if \mathcal{D}' is diagonal. By carrying out explicitly the above calculation we arrive at:

$$\mathcal{D}' = \begin{pmatrix} \mathcal{D}_A \mathcal{A}^2 + \mathcal{D}_B \mathcal{B}^2 & \mathcal{D}_A \mathcal{A} \mathcal{C} + \mathcal{D}_B \mathcal{B} \mathcal{D} \\ \mathcal{D}_A \mathcal{A} \mathcal{C} + \mathcal{D}_B \mathcal{B} \mathcal{D} & \mathcal{D}_A \mathcal{C}^2 + \mathcal{D}_B \mathcal{D}^2 \end{pmatrix}\tag{S-154}$$

Since all matrices involved in the above expression are diagonal by definition, the diagonality condition reduces to:

$$\begin{aligned}\mathcal{D}_A \mathcal{A} \mathcal{C} + \mathcal{D}_B \mathcal{B} \mathcal{D} &= 0 \\ \Leftrightarrow \forall j : (\mathcal{D}_A)_{jj} a_j c_j + (\mathcal{D}_B)_{jj} b_j d_j &= 0\end{aligned}\tag{S-155}$$

We have now established a condition for transforming the Laplacian in a way that mixed derivatives disappear in the new coordinates. This is a generalization of the condition already mentioned in section S3.1. However note that we still can choose the transform coefficients freely as long as the above equation is fulfilled. We will now determine a specific choice that is appropriate for the considered direct binding scenario.

Particular transform for the direct binding scenario

In direct binding one of the particles (A) diffuses on a planar 2D submanifold of the \mathbb{R}^3 while the other particle (B) performs a standard isotropic 3D diffusion in \mathbb{R}^3 . Let us assume that the 2D plane corresponds to the xy -plane of the Cartesian coordinate system, implying $D_{Az} \equiv (\mathcal{D}_A)_{33} = 0$. The diffusion matrix in the original coordinates $\{\mathbf{r}_A, \mathbf{r}_B\}$ then reads:

$$\mathcal{D} = \begin{pmatrix} D_A & 0 & 0 & 0 & 0 & 0 \\ 0 & D_A & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & D_B & 0 & 0 \\ 0 & 0 & 0 & 0 & D_B & 0 \\ 0 & 0 & 0 & 0 & 0 & D_B \end{pmatrix}\tag{S-156}$$

We now specify the coefficients of the transformation matrix \mathcal{M} such that (S-155) holds and the off-diagonal elements of \mathcal{D}' vanish. The latter requires either $b_3 = 0$ or $d_3 = 0$ because of $D_{Az} = 0$ and $D_{Bz} \neq 0$. To ensure that the z -component of

⁸We denote the new representation of the operator with a different sign, but formally $\mathfrak{L}^2 = \mathcal{L}^2$.

the generalized interparticle vector \mathbf{r} is nonzero, as in previous definitions, we opt for $d_3 = 0$. The transformed diffusion matrix $\mathcal{D}' = \mathcal{M}\mathcal{D}\mathcal{M}^T$ then becomes:

$$\mathcal{D}' = \begin{pmatrix} D_A^2 a_1^2 + D_B^2 b_1^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & D_A^2 a_2^2 + D_B^2 b_2^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & D_B b_3^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & D_A^2 c_1^2 + D_B^2 d_1^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & D_A^2 c_2^2 + D_B^2 d_2^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (\text{S-157})$$

With this we may rewrite the Laplacian as follows:

$$\begin{aligned} \mathfrak{L}^2 &= \nabla_Y^T (\mathcal{M}^T \mathcal{D} \mathcal{M}) \nabla_Y \\ &= (D_A a_1^2 + D_B b_1^2) \partial_{r_1}^2 + (D_A a_2^2 + D_B b_2^2) \partial_{r_2}^2 + D_B b_3^2 \partial_{r_3}^2 \\ &\quad + (D_A c_1^2 + D_B d_1^2) \partial_{R_1}^2 + (D_A c_2^2 + D_B d_2^2) \partial_{R_2}^2 \end{aligned} \quad (\text{S-158})$$

The fact that the prefactor of $\partial_{r_3}^2$ is different from the prefactors of the other two components prevents us from regrouping the separate differential operators into a closed form. This is precisely the signature of anisotropic diffusion. Provided that we do not change the (full) rank of \mathcal{M} we may choose the yet undetermined coefficients freely. Here we set

$$a_{1,2} = -1 \quad b_{1,2} = +1 \quad (\text{S-159})$$

to ensure that the first two components of \mathbf{r} reproduce the ones of the standard interparticle vector. Then from (S-158) it is evident that setting $b_3 = \pm \sqrt{\frac{D_A + D_B}{D_B}}$ yields equal prefactors for all three derivatives. With this choice we get rid of diffusion anisotropy by adequately rescaling the r_z -coordinate.

The corresponding transformation matrix \mathcal{M}_{DB} has the form

$$\mathcal{M}_{DB} = \begin{pmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & a_3 & 0 & 0 & \pm \sqrt{1 + \frac{D_A}{D_B}} \\ c_1 & 0 & 0 & d_1 & 0 & 0 \\ 0 & c_2 & 0 & 0 & d_2 & 0 \\ 0 & 0 & c_3 & 0 & 0 & 0 \end{pmatrix} \quad (\text{S-160})$$

and in this particular case the diagonality condition (S-155) reads:

$$\begin{aligned} -D_A c_j + D_B d_j &= 0, \quad j = 1, 2 \\ \underbrace{(D_A)_{33} a_3 c_3}_{=0} + D_B b_3 \underbrace{d_3}_{=0} &= 0 \end{aligned} \quad (\text{S-161})$$

Evidently, the second line is fulfilled for any choice of a_3 , b_3 and c_3 . However, preserving full rank requires $c_3 \neq 0$. An adequate choice is $c_3 = \frac{1}{D_A + D_B}$. Moreover it is convenient to set $a_3 = -b_3$ and

$$c_j = \frac{D_B}{D_A + D_B}, \quad d_j = \frac{D_A}{D_A + D_B}, \quad j = 1, 2 \quad (\text{S-162})$$

With this the particular transform is completely determined, and we finally arrive at the Laplacian in transformed coordinates:

$$\mathfrak{L}_{DB}^2 = \underbrace{(D_A + D_B)}_{\equiv D_r} \nabla_r^2 + \underbrace{\left(\frac{D_A D_B}{D_A + D_B} \right)}_{\equiv D_R} (\partial_{R_1}^2 + \partial_{R_2}^2) \quad (\text{S-163})$$

This is structurally analogous to the operator yielded by the coordinate transform defined in section S3.3. However, here the transformed coordinates are different. The explicit (forward) transformation rules read:

$$\mathbf{r} = \begin{pmatrix} r_{B_1} - r_{A_1} \\ r_{B_2} - r_{A_2} \\ b_3 r_{B_3} + a_3 r_{A_3} \end{pmatrix} = \begin{pmatrix} r_{B_1} - r_{A_1} \\ r_{B_2} - r_{A_2} \\ \varepsilon(r_{B_3} - r_{A_3}) \end{pmatrix}, \quad \varepsilon \equiv \pm \sqrt{1 + \frac{D_A}{D_B}}$$

$$\mathbf{R} = \frac{1}{D_A + D_B} \begin{pmatrix} D_B r_{A_1} + D_A r_{B_1} \\ D_B r_{A_2} + D_A r_{B_2} \\ r_{A_3} \end{pmatrix} \quad (\text{S-164})$$

The formula for \mathbf{r} demonstrates that in the new coordinate system anisotropy is cancelled by rescaling the z -component of the interparticle vector. The sign of the scaling factor ε may be chosen freely; in the following we opt for the positive solution. Note that $r_{A_3} = \text{const}$ because the 2D particle (A) is always in the plane by definition.

For completeness we once again explicitly state the final version of the transformation matrix \mathcal{M}_{DB} :

$$\mathcal{M}_{DB} = \begin{pmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -\varepsilon & 0 & 0 & \varepsilon \\ \varepsilon^{-2} & 0 & 0 & \frac{D_A}{D_B}\varepsilon^{-2} & 0 & 0 \\ 0 & \varepsilon^{-2} & 0 & 0 & \frac{D_A}{D_B}\varepsilon^{-2} & 0 \\ 0 & 0 & D_r^{-1} & 0 & 0 & 0 \end{pmatrix} \quad (\text{S-165})$$

The determinant of this matrix is

$$\det(\mathcal{M}_{DB}) = \frac{-\varepsilon}{D_A + D_B} = \frac{\mp 1}{\sqrt{D_B(D_A + D_B)}} \neq 0 \quad (\text{S-166})$$

confirming that our specific coefficient choice preserves the full-rank property.

Inverse transform

To obtain the inverse transformation rule we simply calculate the inverse of matrix \mathcal{M}_{DB} (S-165):

$$\mathcal{M}_{DB}^{-1} = \begin{pmatrix} -\frac{D_A}{D_B}\varepsilon^{-2} & 0 & 0 & 1 & 0 & 0 \\ 0 & -\frac{D_A}{D_B}\varepsilon^{-2} & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & D_r \\ \varepsilon^{-2} & 0 & 0 & 1 & 0 & 0 \\ 0 & \varepsilon^{-2} & 0 & 0 & 1 & 0 \\ 0 & 0 & \varepsilon^{-1} & 0 & 0 & D_r \end{pmatrix} \quad (\text{S-167})$$

This results in the following back-transform rules:

$$\mathbf{r}_A = \begin{pmatrix} R_1 - \frac{D_A}{D_A + D_B} r_1 \\ R_2 - \frac{D_A}{D_A + D_B} r_2 \\ (D_A + D_B) R_3 \end{pmatrix}$$

$$\mathbf{r}_B = \begin{pmatrix} R_1 + \frac{D_B}{D_A + D_B} r_1 \\ R_2 + \frac{D_B}{D_A + D_B} r_2 \\ (D_A + D_B) R_3 + \sqrt{\frac{D_B}{D_A + D_B}} r_3 \end{pmatrix} \quad (\text{S-168})$$

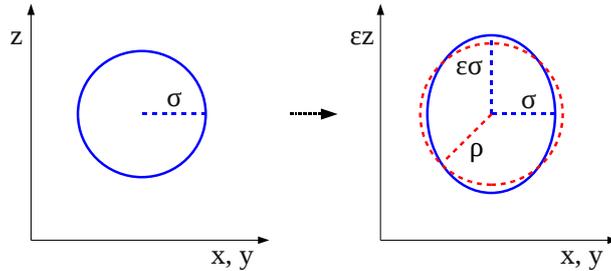


Figure S5: Distorsion of a spherical boundary due to the anisotropic coordinate transform.

We will now explain how the derived transform may be used to sample next-event information for the direct binding scenario using some of the Green's functions that we have already presented within this thesis.

Using known Green's Functions for the transformed problem

By applying the coordinate transform in the way described we succeeded in transforming the two-particle problem in 3D into two separate diffusion problems, namely a 2D-diffusion of the center-of-mass vector \mathbf{R} and a 3D-diffusion of the interparticle vector \mathbf{r} , with a z -axis rescaled by $\varepsilon \geq 1$. Since diffusion of \mathbf{R} in the plane is still isotropic after the transform, we may sample next-event times and new positions for \mathbf{R} in the same way as for the *Planar Surface Pair* (see sec. 2.2.2 in the main text), i.e. by imposing a circular absorbing boundary at $|\mathbf{R}| = R_{\max}$ and reusing the Green's function presented in section S4.3.

The situation is different for the rescaled interparticle vector \mathbf{r} . In 3D, where diffusion of the interparticle vector is isotropic, radiating or absorbing boundary conditions are defined on spheres. Here, by rescaling the z -axis all lengths in z -direction become slightly longer with respect to the other directions in the new coordinate system, meaning that boundaries originally represented by spheres now become prolate spheroids (Figure S5). Since it is technically challenging to compute the Green's function for such boundary conditions, we opted for a simpler, approximative approach, in which the prolate spheroidal boundaries in the transformed coordinates are substituted by spherical boundaries. As an evident advantage, with spherical boundaries we may reuse the well-known 3D Green's function for a radiating inner and absorbing outer boundary. Given that the diffusion constant in the plane is significantly smaller than the diffusion constant in the bulk, e.g. $D_A \simeq D_B/10$, the scaling factor $\varepsilon = \sqrt{1 + D_A/D_B}$ is rather small ($\varepsilon \simeq 1.05$), implying only a minor error induced by the substitution.

As a further modification, we choose the radius of the inner, radiating sphere in such a way that its surface area equals the surface area of the prolate spheroidal. The rationale here is that equal surface areas will ensure that the total probability flux through the new, spherical boundary will be approximately equal to the total flux through the original, prolate boundary. Let the radius of the boundary sphere in the untransformed coordinate system be σ . The sphere transforms to a prolate spheroidal with semi-major axis length $A = \varepsilon\sigma$, whereas the semi-minor axis a is identical to the radius σ (Figure S5). The surface area of a prolate spheroidal is given by:

$$A_P(a, A) = 2\pi \left[a^2 + \frac{aA \arccos\left(\frac{a}{A}\right)}{\sin\left(\arccos\left(\frac{a}{A}\right)\right)} \right] \quad (\text{S-169})$$

Setting this equal to the surface of a sphere with radius ρ , with the particular values

for the half-axes from above, we obtain:

$$4\pi\rho^2 = 2\pi \left[\sigma^2 + \frac{\sigma(\varepsilon\sigma) \arccos\left(\frac{\sigma}{\varepsilon\sigma}\right)}{\sin\left(\arccos\left(\frac{\sigma}{\varepsilon\sigma}\right)\right)} \right]$$

$$\rho = \sqrt{\frac{1}{2} \left(1 + \frac{\varepsilon \arccos\left(\frac{1}{\varepsilon}\right)}{\sin\left(\arccos\left(\frac{1}{\varepsilon}\right)\right)} \right)} \cdot \sigma \quad (\text{S-170})$$

To facilitate calculations we sample next-event information for the transformed inter-particle vector from the regular Green's function that assumes a spherical boundary, using ρ as defined above for the contact radius. Note that after back-transform sampled positions are located on oblate spheroids. This may cause particle overlap when the sampled new distance between the particles is short (comparable to σ); in this case particles are slightly moved apart, the error of which again is small.

S6 Domain and shell construction rules

S6.1 Principal domain making strategy

The governing principle in constructing and sizing of domains in eGFRD is to minimize the computational cost associated with this process. Domain making consists of the following successive steps:

1. Determine which type of domain to construct.
2. Determine the available space for the domain (i.e. its shell).
3. Construct the domain with an optimally sized shell.
4. Draw the next-event time and type for the newly constructed domain.
5. Re-schedule the domain in the central scheduler.

While the cost of the first and last step is roughly the same for all domains, it may vary among different domain types for the other steps. In general, cylindrical domains are more expensive to construct than spherical domains because of the increased computational effort for scaling up cylinders within a specific configuration of other cylinders and spheres. Similarly, *Pair* domains are more expensive than *Single* domains because they require an additional coordinate transform and employ Green's functions which are mathematically more complex.

It is unfeasible to foresee all possible configurations that may occur during eGFRD simulations as required for a real quantitative optimization of the domain making rules. The strategy in defining a unique set of functional rules therefore is to minimize the likelihood of situations that lead to the waste of computational cost, such as repeated reconstruction of domains and construction of expensive domains when it is not strictly advantageous.

S6.2 Social upsizing prevents premature and mutual bursting

Particular care must be put into determining the optimal size of the domain. In principle we want to construct domains as big as possible because their next-event time directly correlates with their size. However, when we size up a large domain such that it will protrude into the direct vicinity of a very small domain, the latter will most likely be updated long before the next event time of the freshly constructed large domain. This may induce premature bursting of the large domain, which then has to be reconstructed from an almost identical situation as before after insignificant time progress, wasting the initial investment of domain making cost. Therefore domains should not be sized up to the maximal available space in any given situation but in a "social" manner, i.e. leave some space for their neighbour domains to avoid very small domains in their own direct vicinity.

In particular, it is important to prevent repeated mutual bursting of two newly constructed adjacent *Single* domains, which may even result in an infinite cycle of domain (re)construction and bursting. Repeated mutual bursting can occur when two particles are at a small distance, yet sufficiently far away to enable the construction of two *Single* domains, and the formation of a *Pair* is disallowed for other reasons (e.g. presence of obstacles). Since domains are sized up in a successive order, using the maximally available space for the first *Single* domain (A) would result in a very small size of the second (B), causing immediate update of B with negligible particle displacement. This in turn would force bursting of domain A in order to size up the B domain again, which would restart the whole process all over from B. This example demonstrates that maximizing domain size is not the same as optimizing it.

S6.3 A minimal Single domain size controls switching to Brownian Dynamics

In section S2 we explain why it is necessary to switch from eGFRD to Brownian Dynamics when propagating particles under crowded conditions. Yet, it is not a priori clear when this switch should be performed. In principle it should be done when the computational cost for *Single* construction divided by the maximal displacement within the domain, which is correlated to domain size, becomes larger than the cost of sampling a trajectory covering the same distance with Brownian Dynamics. Since in GFRD *Single* construction cost is variable, it is hard to devise a general rule. Nevertheless, it is clear that a minimal *Single* has to be defined for proper working of the algorithm. We decided to make this a simulation parameter, the details of which will be described in more detail in the following text.

S6.4 Summarized domain making objectives

The abovementioned general reasoning can be summarized into the following set of simple objectives for efficient domain making rules:

- Construct *Pair* and *Interaction* domains only when interaction is likely, i.e. when particles are close to other particles or reactive surfaces.
- Construct domains socially, i.e. reserve some space for neighbouring domains in order to prevent premature or mutual bursting.
- Construct *Multi* domains (i.e. fall back into Brownian dynamics) if construction of a minimal-size *Single* is impossible.

In order to transform these rules into an applicable algorithm we introduced two length factors which will determine when to start constructing *Pairs* or *Interactions* and when to go into the Brownian dynamics mode during runtime.

S6.5 Two length factors balance the domain making behaviour

Let us define the following two dimensionless constants:

- $\beta \equiv$ “single-shell factor”
- $\mu \equiv$ “multi-shell factor”

For a given particle radius R these two factors define the half-size βR of the minimal *Single* shell and the radius μR of the (always spherical) *Multi* shell of that particle, respectively. Note that the minimal *Single* shell can be either a sphere or a cylinder, depending on whether the particle is a 3D, 2D or 1D species. The requirements $\beta \geq 1$ and $\mu \geq 1$ are obvious. Since a *Multi* shell shall never be constructed when there is enough space for a minimal *Single* shell, we also require $\beta > \mu$.

Let R_0 be the radius of a particle P_0 for which we want to construct a domain, R_1 the radius of its nearest neighbour particle P_1 and $\sigma \equiv R_0 + R_1$. Then, based on β and μ , we define the following lengths for P_0 :

- the “reaction horizon” $\equiv \beta R_0$
- the “multi horizon” $\equiv \mu R_0$

and, as specializations of the above:

- the “pair horizon” $\equiv \beta \sigma$
- the “surface horizon” \equiv “reaction horizon” \equiv “burst horizon/radius” $= \beta R_0$
- the “multi-partner horizon” $\equiv \mu \sigma$

Different naming highlights different purpose for these of the above quantities that otherwise are equal to each other.

The reaction horizon is used to determine when a *Pair* or *Interaction* domain should be constructed instead of a *Single* domain. The multi horizon defines when *Single* construction should be dropped in favor of *Multi* construction. The pair horizon and surface horizon are specifications of the reaction horizon for *Pair* and *Interaction Single* formation: while an *Interaction* is formed when a surface is within the reaction horizon, a *Pair* construction is attempted only when the reaction horizons of the two involved particles overlap, i.e. when the center of mass of P_1 is within the pair horizon of P_0 . Similarly, the algorithm will switch into Brownian Dynamics mode when a surface is within the multi horizon of P_0 or when the multi horizons of P_0 and P_1 overlap, i.e. when P_1 is within the multi-partner horizon of P_0 .

The burst horizon defines the volume within which a particle will burst neighbouring domains. Since the objective of bursting is to generate space for at least a minimal *Single* shell the burst horizon should be at least as big as the reaction horizon. Since there is no evident necessity to make it bigger than the reaction horizon, we conveniently set these lengths to be equal.

Practically β and μ can be used to tune the overall behaviour and performance of domain making: Increasing μ will prompt the simulation to switch to Brownian Dynamics earlier. Whether this is advantageous or not depends the crossover radius at which construction of small *Single* domains yields a smaller average simulation time advance per computational cost unit than the construction of *Multi* domains. In a similar way, whether a larger or smaller β is favorable depends on the average likelihood of reactions in the system. Since the latter depend on the parameters, there is no obviously optimal choice for β and μ . We find that $\beta \in [2, 3]$ and $\mu \in [\sqrt{3}, 2]$ gives reasonable performance.

S6.6 Domain making algorithm

We can now compile the abovementioned postulations and definitions into a well-defined algorithm for domain making. Let us imagine a particle which just exited from whatever domain type after an update. An update can be either triggered by a next-event picked from the scheduler (i.e. a reaction, domain exit or surface interaction) or by premature bursting of neighbouring particles induced in the aftermath of such scheduler event. The particle by default is put into the system as a *Non-Interaction Single* with a shell that just envelopes the particle. Note that this “zero-shell” is spherical for 3D particles and cylindrical for 2D and 1D particles. We will call a *Single* with a zero-shell a “*Zero-Single*”. Each *Zero-Single* is put into the scheduler with zero next-event time ($dt = 0$) in order to reconstruct its domain immediately after it was produced.

We then perform the following order of actions to construct a new domain:

1. **Bursting:** Burst any neighbouring “intruder”, i.e. a domain that intrudes into the burst radius of the particle, with the exception of *Multi* domains and other *Zero-Singles*, i.e. domains which are yet to pass through the domain making procedure themselves. By default the burst radius is equal to the reaction horizon. Burst recursively, i.e. whenever a bursted intruder has intruders within its own reaction horizon, also burst these. The following steps then are repeated for each *Zero-Single* present in the system after bursting.
2. **Reaction/interaction attempt:** Compile a list of all potential interaction partners (particles or reactive surfaces). Pick the closest interaction partner and try a reaction (with particles) or interaction (with surfaces) if the closest partner is within the specified reaction or interaction horizon. If a minimal reaction

(*Pair*) domain or *Interaction* domain can be constructed, size it up socially to the maximal available space and go directly to step (4.).

3. **Single domain upsizing attempt:** If a *Pair* or *Interaction* could not be constructed, yet the closest partner is within the multi horizon of the *Non-Interaction Single*, then (recursively) construct a *Multi* domain (as specified further below) and proceed directly to step (4.). Else, size up the *Non-Interaction Single* domain socially to the maximally available space and continue.
4. **Re-scheduling:** For the constructed domain type determine the next-event time and type and reinsert this information into the scheduler.
5. **Repeat** the whole procedure for the next *Zero-Single* until there are no more *Zero-Singles* in the scheduler.

The pseudo-code of the domain making algorithm is shown in box “Algorithm 2”.

Algorithm 2 The eGFRD domain making algorithm.

```

Z ← Zero-Single
while z ∈ Z do

  for all domain in burst radius of z do
    if domain is not Multi and dt(domain) > 0 then
      Z_bursted ← burst domain recursively
      Z ∪ Z_bursted
    end if
  end for

  S ← {neighbouring surfaces of z}
  P ← {neighbouring particles of z}
  c ← closest object n ∈ S ∪ P

  if c ∈ S and c in surface horizon of z then
    successful ← try interaction of z with c
  else if not successful and c ∈ P and c in pair horizon of z then
    successful ← try to form Pair (z, c)
  else if not successful and c out of multi horizon of z then
    successful ← try to scale up shell of z
  else if not successful then
    form Multi from z with c recursively
  end if

  re-schedule z
  remove z from Z

end while

```

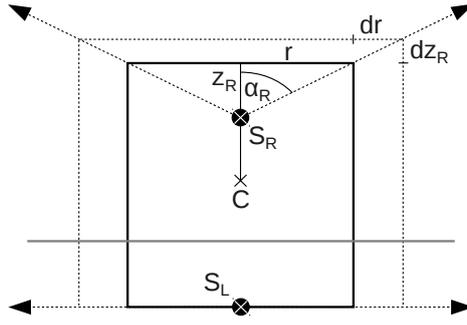


Figure S6: Cylinder scaling in eGFRD. The cylinder is scaled differently on its two sides (L and R) from two separate scale centers S_L and S_R which here do not coincide with the center point (C). For each side, the scale angle (α_L , α_R) defines the aspect ratio at scaling, i.e. how the respective height (z_L , z_R) scales with the cylinder radius (r). In the given example, $\alpha_L = \pi/2$, meaning that $z_L (= 0)$ remains constant upon changing r .

S6.7 Multi construction

In eGFRD *Multis* are contingent three-dimensional objects made up from either one spherical *Multi* shell or a set of overlapping spherical *Multi* shells. The radius of a *Multi* shell is equal to the particle radius plus the reaction length multiplied by the multi-shell factor $\mu > 1$. *Multi* domains are constructed recursively: When a *Zero-Single* z has been determined to form a *Multi* object it checks for objects within its surroundings. Any other *Zero-Single* z' that is within the common multi horizon $\mu(R_z + R_{z'})$ will be added to the *Multi*. Then, for each z' that was added, the same check is performed for its surroundings, ignoring z . This is repeated until no further *Zero-Singles* can be added to the *Multi* object. Note that *Multi* shells in such configurations in principle can overlap with more than one other *Multi* shells. If there are only surfaces within the horizon the *Multi* will consist of only one *Multi* shell, the one around z .

S6.8 The test-shell concept

In order to prevent wasted effort, in eGFRD the sizing of a domain shell upon domain (re)construction is decoupled from sampling of next-event information from the Green's function. This is achieved by using "test shells". In a particular situation in which a new domain has to be created, the simulator first attempts to determine the maximal size of the tentative test shell of the domain, taking into account the required shell geometry (cylindrical or spherical) and particular scaling parameters (e.g. the scale aspect ratio of cylinders). Starting from a (predefined) maximal shell size, the test shell then is scaled down successively with respect to each neighboring shell via collision detection. During the collision detection step the maximal dimensions of the test shell that does not lead to an overlap with the particular other domain shell are calculated (see subsequent section for more detail). If at the end of the scaling procedure the dimensions of the test shell are not smaller than the required minimal dimensions (determined by the factors β and μ defined further above and particle radii) a new domain object is parametrized with the test shell, and only after this step its next-event information is sampled. In the opposite case the construction of the respective domain type is rejected and the algorithm proceeds by attempting the construction of another domain type (e.g. *Single* or *Multi*).

S6.9 Shell collision detection

Considerable computational effort has to be put into detection of collisions between a scaled test shell and another (static) shell. While this problem is trivial when scaling spheres or cylinders against spheres or parallel cylinders against each other, it is, maybe surprisingly, less straightforward for arbitrarily oriented and even orthogonal cylinders. Note that in eGFRD cylinder scaling is performed subject to a fixed (but in principle arbitrary) “scale aspect ratio” (defining a certain “scale angle”), which links the change in cylinder height to the change in the radius. The aspect ratio is usually set by the requirement of equalizing expected first passage times of the enclosed particle towards the cap and the tube of the cylindrical shell; as illustrated by Figure S6, the scale angle may differ for the two opposite sides of the cylinder⁹. In general, also the reference points of the scaling (“scale centers”) do not coincide with the midpoint of the cylinder, particularly in cases in which scale angles are indeed different on both sides. Thus, scaling a cylindrical shell in eGFRD conceptually consists of scaling the two sides of the cylinder separately, albeit linked via the common radius.

To scale orthogonal cylinders against each other in eGFRD we consider the two-cylinder problem in a standardized cartesian coordinate system centered at the midpoint of the scaled cylinder, with x-base-vector pointing towards the midpoint of the static cylinder and z-base-vector coinciding with the axis of the scaled cylinder. We then determine the specific type of collision that may occur upon scaling up the cylindrical test shell. There are seven possible collision types:

1. TF: the tube of the scaled cylinder hits the flat side of the static cylinder.
2. TT: the tube of the scaled cylinder hits the tube of the static cylinder.
3. TE: the tube of the scaled cylinder hits the edge¹⁰ of the static cylinder.
4. FT: the flat side of the scaled cylinder hits the tube of the static cylinder.
5. ET: the edge of the scaled cylinder hits the tube of the static cylinder.
6. EE: the edge of the scaled cylinder hits the edge of the static cylinder.
7. None: no collision possible in the given scenario¹¹.

Identification of the collision type is facilitated by comparing the location of the projected midpoint of the scaled cylinder to the projected edges of the static cylinder in the xy-plane of the standardized coordinate system (in which the scaled cylinder appears circular and the static cylinder rectangular). Certain respective locations exclude certain collision types; for example, if the midpoint of the scaled cylinder is within the rectangular projection of the static cylinder, the collision must be of type FT (given that the height is scaled). Once the collision type is known, the maximal dimensions of the scaled cylinder are determined taking into account its “intrinsic” scaling properties (scale angle, location of scale center, minimal size). This is mostly achieved via straightforward geometric calculations. For collision type EE a closed form for the new dimensions could not be obtained; the values are therefore calculated from an implicit equation using a numerical rootfinder. The detailed calculations are part of the eGFRD technical documentation and beyond the scope of this thesis.

A concise scheme for scaling arbitrarily oriented eGFRD-type cylinders against each other is yet to be devised.

⁹This is e.g. the case for the shell of the *Planar Surface Interaction* domain (see section 2.2.1), the height of which is scaled only on the side of the planar surface facing the particle, while the height on the opposite side is kept fixed when the radius is scaled. This is the example shown in Fig. S6, where the gray line represents a planar surface.

¹⁰defined as the circular line that separates cylinder tube and flat side

¹¹This e.g. may occur in cases in which only the radius or height are scaled.

S6.10 Convergence issues affecting domain construction

The Green's functions, survival probabilities, cumulative PDFs and the expressions for the boundary fluxes used in eGFRD typically have the common form

$$C \cdot \sum_{n=0}^{\infty} e^{-\zeta_n^2 D t} X_n \quad (\text{S-171})$$

where C is constant and X_n does not depend on t . It can be shown that ζ_n^2 , to a good approximation, scales linearly with n and inversely with the domain size L , while typically $|X_n| \sim 1$. Convergence of these sums thus is dominated by the exponential terms. We found that it is severely hampered for evaluation times t which are small on the typical timescale of the domain, i.e. the mean time required to traverse it by diffusion with diffusion constant D . This is the case when the distance Δ of the particle to the closest boundary becomes very small. Then evaluation times are of the order of $t_\Delta = \Delta^2/2dD$, where d is the dimensionality. A fair estimate for the number n of summation terms needed to reach a desired convergence threshold ε follows from:

$$e^{-(\frac{cn}{L})^2 D t} < \varepsilon \quad \Leftrightarrow \quad n > \frac{L\sqrt{1/\varepsilon}}{c\sqrt{Dt}} \quad (\text{S-172})$$

where we approximate $\zeta_n = cn/L$ with $c = \text{const}$. Inserting t_Δ into the above equation yields

$$n_\Delta > \frac{L}{\Delta} \cdot \text{const} \quad (\text{S-173})$$

showing that the required number of terms to reach a predefined convergence accuracy scales inversely with the distance Δ to the closest boundary. Therefore it should be avoided to construct domains in a way that Δ/L is small; optimally domains should be constructed in a way that distances between the starting point of the diffusing particle and the domain boundaries are approximately equal.

In practice this is handled in two ways: Whenever particles start very close to radiating or absorbing boundaries we construct a domain that does not use all available space but is only sized up to $L \simeq 2\Delta$. While this requires the successive creation of undersized domains (resulting in minor next-event times) in order to elongate the distance between the reactive boundary and the particle, it overcomes the above-mentioned convergence issues by keeping Δ/L constant. Alternatively, we scale domains up as much as possible and, where available, employ Green's functions that are bounded unilaterally, i.e. neglect the distant second boundary. These Green's functions typically are finite sums, which facilitates their implementation and computation, and in the above case approximate the double-bounded solutions very well.

S7 Code profiling

In order to assess the computational performance of our new eGFRD version we recorded the CPU time per simulated (“real”) time as a function of the (effective) number of simulated particles at constant volume, for the following scenarios:

- “3d, nR”: N identical particles diffusing in a box (“cytoplasm”) with dimensions $L \times L \times L$ and periodic boundary conditions in x -, y - and z -directions.
- “3d, wR”: The same setup, but with initially N particles of species A engaging in the simple reversible binding reaction $2A \rightleftharpoons B$, where B is another cytoplasmic species that differs from A only by its label and reactive behavior.
- “2d, nR”: N identical particles diffusing on a plane (“membrane”) with dimensions $L \times L$ and periodic boundary conditions in x - and y -directions.
- “2d, wR”: The same setup, but with initially N particles of species A engaging in the simple reversible binding reaction $2A \rightleftharpoons B$, where B is another plane-bound species that differs from A only by its label and reactive behavior.
- “1d, nR”: N identical particles diffusing and drifting on a cylinder (“microtubule”) of length L and periodic boundary conditions in x -direction.
- “1d, wR”: The same setup, but with initially N particles of species A engaging in the simple reversible binding reaction $2A \rightleftharpoons B$, where B is another cylinder-bound species that differs from A only by its label and reactive behavior.

The parameters used for the respective cases are given in Table S1. We chose the diffusion coefficients such that they correspond to realistic values in biological systems, and reaction parameters fast enough as to ensure that both the forward and backward reactions actually occur sufficiently often on the simulated time scale. To this end, we chose a fast dissociation rate $k_b = 100/s$ in all dimensions, while the forward rates were set to multiples of the predicted diffusion-limited rates in the respective dimensions; in the lower dimensions the latter are density-dependent. Note that the drift in the 1d cases is only included in order to comprise all new features in the profiling; due to the periodic boundary conditions, it does not alter the expected spatial distribution on the cylinder.

For each setting and each set of parameters, we first carried out a calibration run by propagating 10 independent samples towards a stationary state in which the average copy numbers only fluctuate insignificantly between subsequent runs; this was followed by a profiling run in which the 10 samples were simulated further for a fixed amount of steps ($N_p = 10^7$) while the CPU time and the (additional) simulated time were recorded. In order to obtain a reference for comparison, the profiling run was also carried out from the same initial conditions using our rvm-BD scheme (see sec. S2.2) for particle propagation exclusively instead of eGFRD (which occasionally also switches to rvm-BD, but only in very crowded situations).

The results for the respective cases above are summarized in Fig. S7. The figure demonstrates that at low copy numbers ($\lesssim 100 - 1000$ for the given system size, translating to $\lesssim \mu M$ concentrations in 3d), the extended eGFRD scheme is up to 3 orders of magnitude faster than our (optimized) rvm-BD scheme. In 2d and 3d the advantage of eGFRD over rvm-BD is less pronounced than in 3d, and the crossover at which rvm-BD becomes more efficient occurs at lower copy numbers, as expected from the fact that in lower dimensions the same amount of particles get crowded much faster than in 3d. For the same reason, as explained in more detail in the following Section S7.1, the eGFRD runtimes scale with the particle number with different exponents in the respective dimensions, while the BD runtimes increase (approximately) linearly with the particle number in all dimensions; the recorded data reproduces this

Symbol	Name	Value / Formula	Unit
L	System size	1.0	μm
R_p	Particle radius	5.0	nm
$\Rightarrow \sigma$	Particle contact radius	10.0	nm
$N = N_A$	Number of A particles	varied	
D_1	1d-diffusion constant on cylinder	0.1	$\mu\text{m}^2/\text{s}$
D_2	2d-diffusion constant on plane	0.1	$\mu\text{m}^2/\text{s}$
D_3	3d-diffusion constant in cytoplasm	10.0	$\mu\text{m}^2/\text{s}$
v	drift velocity on microtubules	1.0	$\mu\text{m}/\text{s}$
$k_{f,1}$	intrinsic binding rate in 1d	$2D_1N/L$	m/s
$k_{f,2}$	intrinsic binding in 2d	$10 \cdot 4\pi D_2 / \log(L^2/N\sigma^2)$	m^2/s
$k_{f,3}$	intrinsic binding in 3d	$4\pi\sigma D_1$	m^3/s
k_b	unbinding rate	100	$1/\text{s}$

Table S1: Overview of the standard parameters used in the profiling simulation runs. For the binding rates we use (multiples of) the diffusion-limited binding rates derived by KIVENSON & HAGAN [134], which are different for each respective dimension and density-dependent in 1d and 2d (factors N/L and N/L^2 in the formulas).

scaling in parts of the scanned particle number ranges, with deviations at very low copy numbers and in the crossover regimes; the latter can be attributed to crowding and an overall increased overhead originating from frequent particle interactions, the former from the fact that this overhead is not present at particle numbers ~ 1 .

Fig. S8 shows a comparison between the dimensions, again for the cases without (upper panels) and with reversible reactions (lower panels). In addition to replotting the data already shown in the separate panels of Fig. S7 in the left column of Fig. S8, the respective right column panels display the data with an alternative quantity on the x-axis that allows for a more “fair comparison” at (a hypothetical) equivalent degree of crowding, captured by a “3d density equivalent” defined as

$$\rho_{\text{eqv}}(N_{\text{eff}}) \equiv \frac{N_{\text{eff}}^{3/d}}{L^3} \quad (\text{S-174})$$

for the respective dimension d . For the lower dimensions, the “3d density equivalent” maps the lower-dimension density to a 3d concentration that would result in an average interparticle distance (roughly) equal to the one in the lower dimension; in 3d it is simply identical to the classical definition of the particle concentration. In the right column of Fig. S8 we measure these concentrations in nM units, but we refer to it as “nM equivalent” (nM_{eqv}) to emphasize the hypothetical character of the quantity. The plots in the left column demonstrate that, as expected, the crossover at which eGFRD becomes more expensive than BD happens at smaller particle numbers in 1d and 2d as compared to 3d; however, the “fair comparison” in the right column also shows that the crossover occurs at lower equivalent degree of crowding in 2d than in 3d, but at a markedly higher degree of crowding in 1d.

Overall, although the profiling reveals that our prototype implementation still offers room for improvements, our eGFRD scheme compares well against an optimized, smart version of BD; it should be emphasized that in the rvm-BD scheme that we compare to, the average propagation time steps are more than 3 orders of magnitude higher than in a conventional brute-force BD scheme with a time step $\Delta t \sim 10^{-6} \sigma^2/D$.

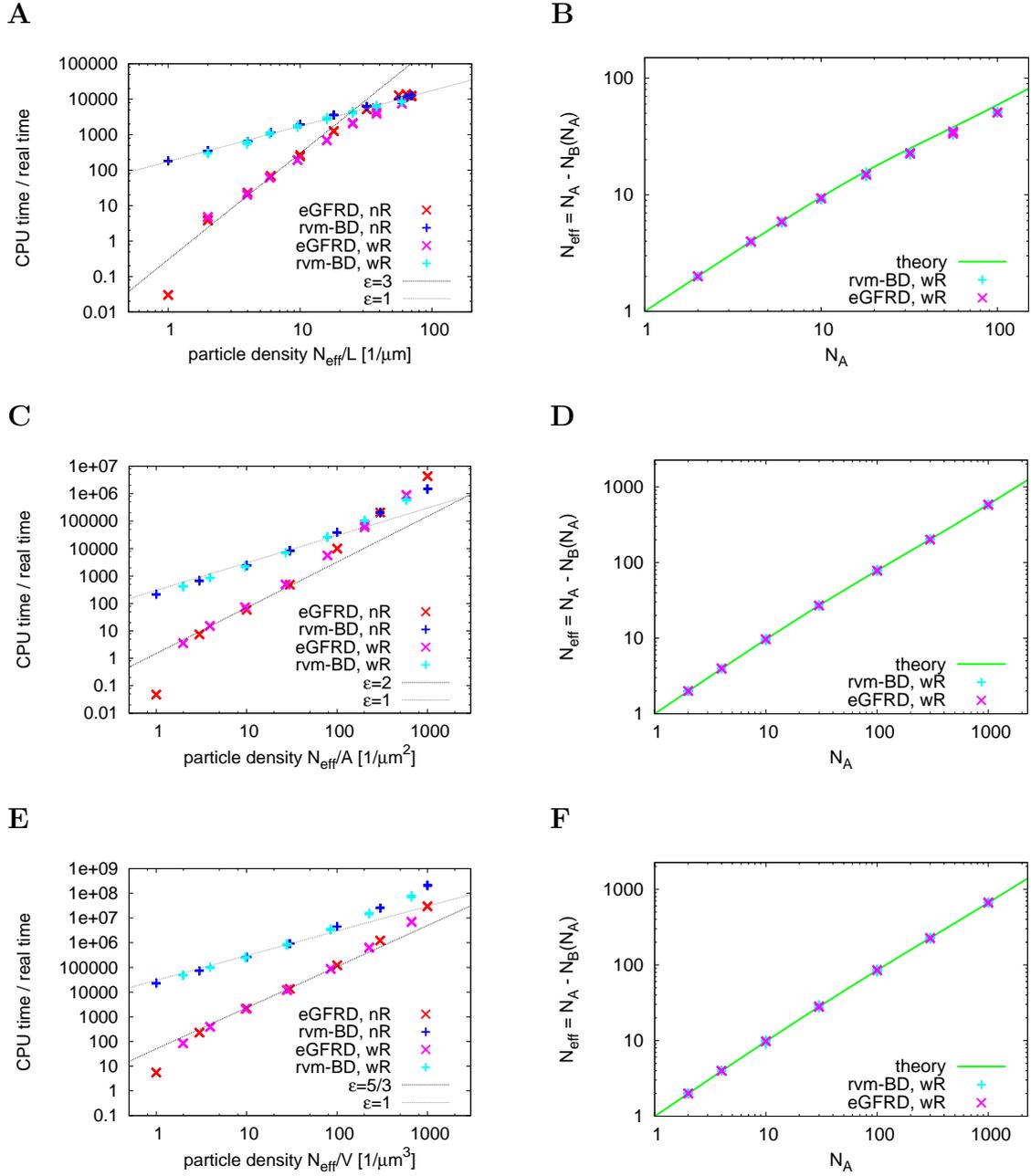


Figure S7: Code profiling results. In the left column we show the CPU time per real (simulated) time as a function of the effective particle density N_{eff}/L^d ($d = 1, 2, 3$), for the cases without reactions (nR) and with a reversible reaction $2A \rightleftharpoons B$ (wR), in 1d (A), 2d (C) and 3d (E); in all cases, the eGFRD runs (red and magenta) are compared to runs carried out via our reaction-volume method BD scheme (rvm-BD, blue and cyan); the black and grey dashed lines show the expected scaling of the CPU time with N_{eff} for eGFRD ($\sim N^{\varepsilon > 1}$, dimension-dependent, see Sec. S7.1) and BD (always $\sim N^1$), respectively. Note that for the “nR” runs $N_{\text{eff}} = N$, while for the “wR” runs the total equilibrium copy number $N_{\text{eff}}(N) \leq N$ is density-dependent; we plot the predicted and measured values of $N_{\text{eff}}(N)$ in the right column, again for 1d (B), 2d (D), and 3d (F). Throughout all plots, each cross shows the data for one individual sample, with a sample size of 10 for each set of parameters.

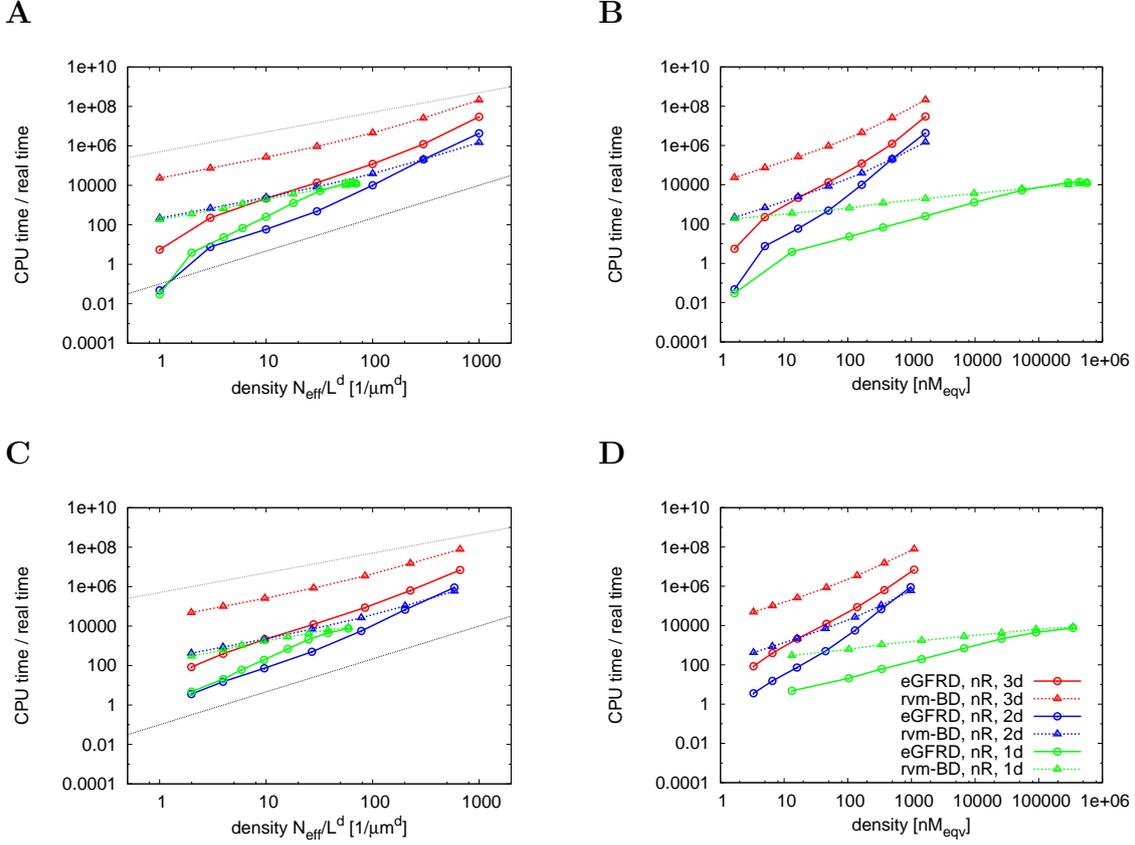


Figure S8: Code profiling comparison. Here we show a comparison of the profiling results in the respective dimensions, for the case without (nR, top row) and with reversible reactions (wR, bottom row). The legend in panel (D) is valid for all panels; the grey dashed line shows the predicted scaling for BD ($\sim N^1$), the black dashed line the predicted scaling for 3d-eGFRD ($\sim N^{5/3}$); the colored solid (eGFRD) and dashed (rvm-BD) lines are guidelines for the eye. The plots in the left column display the CPU-time per simulated time as a function of the density N_{eff}/L^d in the respective dimension d , while the right column shows a “fair comparison” in which we convert the density to a “nM equivalent” (nM_{eqv}) by rescaling the x-axis from N_{eff}/L^d to an effective 3d concentration $N_{\text{eff}}^{3/d}/L^3$ measured in nM; this unit allows a comparison at (a hypothetical) equivalent degree of crowding, or equivalent average interparticle distance. Note that the 100-fold difference in CPU-time per real time in 3d as compared to 2d and 1d is expected, due to the 100-fold difference in the diffusion coefficients used in the respective dimensions.

S7.1 Expected scaling of CPU time with the particle number

While in BD simulations with a fixed time step Δt the computational cost is expected to scale linearly with the number of propagated particles N (given that particle interactions are treated in a smart way, e.g. via cell lists, such that the cost for simulating particle reactions is not disproportionately higher than the cost of their propagation), the following scaling argument can be made for eGFRD: Generally, for a scheme with varying propagation time step Δt , we expect the runtime T to scale in proportion to the particle number N , but inversely with the average time step $\langle \Delta t \rangle$. In eGFRD, $\langle \Delta t \rangle$ directly depends on the space available for diffusion without particle collisions, and therefore is roughly proportional to the square of the average distance between the particles $\langle \lambda \rangle$: $\langle \Delta t \rangle \sim \langle \lambda \rangle^2$. The average interparticle distance $\langle \lambda \rangle$, however, scales differently with N in the respective dimensions: $\langle \lambda \rangle \sim N^{-1/d}$. Taken together this yields

$$T \sim \frac{N}{\langle \Delta t \rangle} \sim \frac{N}{\langle \lambda \rangle^2} \sim N^{1+\frac{2}{d}} \quad (\text{S-175})$$

and for the respective dimensions:

$$T \sim N^{\frac{5}{3}} \quad \text{in 3d} \quad (\text{S-176})$$

$$T \sim N^2 \quad \text{in 2d} \quad (\text{S-177})$$

$$T \sim N^3 \quad \text{in 1d} \quad (\text{S-178})$$

S7.2 Formulas for stationary equilibria in the simulations with reversible reactions

For completeness, here we briefly derive the formulas for the effective copy number in our example reversible-binding reaction $A + A \rightleftharpoons B$, defined as

$$N_{\text{eff}} \equiv \langle N_A \rangle + \langle N_B \rangle \quad , \quad (\text{S-179})$$

where the brackets refer to the stationary mean of the copy numbers for the respective species.

The stationary copy numbers can be obtained by solving the following simple ODE for the particle densities $\rho_X \equiv N_X/S$ (where S is the dimension-dependent, relevant system size)

$$\partial_t \rho_B = k_f \left(\frac{\rho_A}{2} \right)^2 - k_b \rho_B \quad (\text{S-180})$$

in steady state ($\partial_t \rho_B \equiv 0$), under the side constraint that—at any moment—the initial total copy number $N \equiv N_A(0)$ is conserved as follows:

$$N_A(t) + 2N_B(t) \equiv N \quad (\text{S-181})$$

Note that this implies $N_{\text{eff}} = N - \langle N_B \rangle$. Furthermore, since in some of our simulations crowding effects are significant, it is important to use the effective system size reduced by the fraction taken up by the particles for calculating the densities. The steady-state equation to solve then reads:

$$\partial_t \frac{\langle N_B \rangle}{S_0 - N_{\text{eff}} S_p} = \frac{k_f}{4} \left(\frac{N - 2\langle N_B \rangle}{S_0 - N_{\text{eff}} S_p} \right)^2 - k_b \frac{\langle N_B \rangle}{S_0 - N_{\text{eff}} S_p} = 0 \quad (\text{S-182})$$

$$\Leftrightarrow (N - 2\langle N_B \rangle)^2 = 4K_d [S_0 - (N - \langle N_B \rangle) S_p] \langle N_B \rangle \quad (\text{S-183})$$

where $K_d = k_b/k_f$ and S_0 denotes the total available space, while S_p is the space taken up by a single particle; these quantities depend on the dimension as follows (R_p being the particle radius): $S_0 = L^3$ and $S_p = 4/3 \pi R_p^3$ in 3d; $S_0 = L^2$ and $S_p = \pi R_p^2$ in 2d; $S_0 = L$ and $S_p = 2R_p$ in 1d.

This yields the stationary solution

$$\langle N_B \rangle = \frac{N + K_d(S_0 - NS_p) \pm \sqrt{N^2 S_p (K_d^2 S_p - 1) - 2NS_0(K_d^2 S_p - 1) + K_d^2 S_0^2}}{2 - 2K_d S_p} \quad (\text{S-184})$$

where only the solution with the negative sign matches the condition $\langle N_B \rangle = 0$ for $N = 0$. From this we obtain:

$$\begin{aligned} N_{\text{eff}}(N) &= N - \langle N_B \rangle \\ &= \frac{N - K_d(S_0 + NS_p) + \sqrt{N^2 S_p (K_d^2 S_p - 1) - 2NS_0(K_d^2 S_p - 1) + K_d^2 S_0^2}}{2 - 2K_d S_p} \end{aligned} \quad (\text{S-185})$$

This formula is valid in all dimensions, but note that in our profiling simulations $K_d = k_b/k_f$ changes with d ; more specifically, $k_b = 100/s$ is held constant in all dimensions, while for the forward rate we chose multiples of the diffusion-limited rate in the respective dimensions (see Table S1); for the diffusion-limited rates we use the expressions derived by KIVENSON & HAGAN [134].

The average sampled copy numbers in our simulations agree very well the predictions of Eq. (S-185), see Fig. S7, panels B, D and F; only in 1d (panel B) we observe that at higher copy numbers the stationary equilibrium is slightly biased towards the product state, resulting in an overall lower N_{eff} ; we attribute this to the fact that in crowded situations the dissociation reaction is more often rejected, due to lack of space; the fact that both the eGFRD and BD simulations deviate from the theory to the same degree support the notion that the deviation is not an artifact of the simulation methods, but a limitation of the modelling assumptions (i.e., the naive handling of dissociation attempts with insufficient space).

S8 Pom1-gradient simulations

S8.1 Parameter choice

Table S2 contains an overview of the parameters that we used for our exemplary simulations of Pom1 gradient formation, described in detail in section 3 of the main text. Where available, we took parameter values from the literature (sources given); otherwise we employed typical values or own estimates resulting in gradients that (roughly) reproduce the experimental observations, in particular the gradient length scale; more detailed comments on this are found in the table and corresponding footnotes.

S8.2 Data acquisition and analysis

Here we briefly describe how we recorded and analysed our Pom1 gradient data.

Starting from a situation in which all Pom1 particles are randomly placed in the cytoplasm in their fully phosphorylated state, we first propagated the simulations until all of them reached a simulated (real) time $t \geq 30$ s, to allow the Pom1 gradient on the membrane to be established and equilibrate. We then continued the simulations for measurement runs for an additional simulated time of at least 30 s, and recorded the complete particle trajectories (positions and species) with a data acquisition interval of 0.1 s.

After the simulations had terminated, the copy numbers of the membrane-associated Pom1 in the respective phosphoforms, N_n , $n = 1..n_p$, and our main observable, the total number of membrane-bound Pom1 in all phosphoforms, $N_{m,tot} \equiv \sum_{n=1}^{n_p} N_n$, was binned into 2d-histograms of particle density ($\rho_m(x, y) \equiv$ local no. of particles / bin area) with 50×50 bins stretching over the whole membrane plane ($10 \mu\text{m} \times 10 \mu\text{m}$); this was done both for individual time frames and with all measured data in one histogram, whereby in the latter case we also computed the (local) variance with respect to the time average in each histogram bin. The same binning and averaging procedure was also carried out for a one-dimensional histogram in which the abscissa is the radial distance from the gradient origin (particle injection point) at $x_0 = 5 \mu\text{m}$, $y_0 = 5 \mu\text{m}$, i.e. a histogram of $\rho_m(\Delta r)$ with $\Delta r \equiv \sqrt{(x - x_0)^2 + (y - y_0)^2}$.

We then fitted exponential functions of the form $f(\delta) = A \exp(-\delta/L)$ both to the x- and y-sections of the 2d-density histograms (thus with $\delta = (x - x_0)$ and $\delta = (y - x_0)$), and to the 1d-histogram (with $\delta = \Delta r$ as defined above) in order to obtain the gradient amplitude A and the gradient decay length L for the given set of parameters. For the vast majority of gradients the fitted values of A and L do not depend much on which of the fits is used; however, we decided to proceed with the values from the fits to the radial 1d-histogram because they comprise all binned data in one graph.

S8.3 Results for the trans-model with slower phosphorylation rates

In the main text we present our results for a trans-phosphorylation model in which k_u , the unbinding rate of fully phosphorylated Pom1, is varied, while k_{pt} , the rate of in-complex trans-phosphorylation and complex dissociation, is set to a very fast value (1000/s). This is an ad-hoc assumption, and the real phosphorylation rate is unknown. In order to assess how our results depend on this, we also carried out simulations in which we vary k_{pt} at constant unbinding rate $k_u = 5/s$. The results are summarized in Fig. S9. We observe that slow trans-phosphorylation rates ($< 100/s$) lower the maximal gradient amplitude (panel A), while simultaneously the gradient length scale increases (panel B); this is consistent with the fact that slower

Symbol	Name	Value	Unit	R/C
L	Side length of sim. box and membr. plane	10	μm	
L_{cyl}	Length of microtubule cylinder	9.85	μm	
R_{cyl}	Radius of microtubule cylinder	25	nm	
R_{part}	Particle radius	10	nm	
N_{p}	Total Pom1 particle number	480		
D_{cyt}	Pom1 cytoplasmic diff. const.	1.5	$\mu\text{m}^2/\text{s}$	[135]
D_{mem}	membrane-bound Pom1m diff. const.	0.026	$\mu\text{m}^2/\text{s}$	[135]
D_{cyl}	diffusion const. on microtubules	0.05	$\mu\text{m}^2/\text{s}$	
v	drift velocity on microtubules	0.5	$\mu\text{m}/\text{s}$	
λ	Pom1 gradient decay length	1.5 ± 0.4	μm	[135]
n_{p}	no. of Pom1 phosphorylation sites ¹²	6		[106]
$k_{\text{p},n}, n = 1..6$	Pom1m cis-autophosphorylation rate	0.05	1/s	OE ¹³
k_{pt}	Pom1m in-complex phosphorylation-dissociation rate	1000	1/s	OE
$k_{\text{u},n}, n = 1..5$	unbinding rate of n times phosph. Pom1m	0.0	1/s	OE
$k_{\text{u}} \equiv k_{\text{u},6}$	unbinding rate of 6 times phosph. Pom1m	0.05	1/s	OE
$k_{\text{b},2\text{d}}$	intrinsic bdg. rate on membr. and to cylinder	10^{-11}	m^2/s	DL ¹⁴

Table S2: Overview of the parameters used in the simulations of Pom1 gradient formation. In the right-most column we either list the source from which we obtained the given parameter value, or alternatively comment on the motivation for our choice.

release after Pom1-dimer formation allows the particles to diffuse away further from the injection point before they are fully phosphorylated and ready to unbind.

Yet, the observed dependencies of the amplitudes and length scales per se do not tell us how the buffering effect in the trans-phosphorylation model is affected by k_{pt} . This dependency is shown in panel C of Fig. S9. Here we see that the anticorrelation between gradient length and amplitude is preserved as long as $k_{\text{pt}} \geq 10/\text{s}$, although the maximal amplitudes for $k_{\text{pt}} = 10/\text{s}$ are already down to approximately half the values as for faster phosphorylation rates; only for very slow $k_{\text{pt}} = 1/\text{s}$ the slope in the anticorrelation plot is reduced from ~ -0.5 to ~ -0.3 . As evidenced by panel D of Fig. S9, altering k_{pt} does not affect the Poissonian character of the observed variances.

Overall, the buffering effect in the trans-model thus turns out to be very robust with respect to the value of the in-complex phosphorylation-dissociation rate k_{pt} .

S8.4 Results for the cis-model

For comparison, we also repeated the simulations of Pom1 gradient formation for the cis-autophosphorylation model, in which each membrane-bound Pom1 particle can autonomously autophosphorylate itself, without the need to encounter another Pom1 particle. Again, we varied the Pom1 injection rate j and the rate of unbinding upon complete autophosphorylation, k_{u} . The results for the cis-model are presented in Fig. S10. Panels A and B of the figure show that—as long as the injection rate is high enough ($j \gtrsim 3/\text{s}$)—neither the gradient amplitude A nor its length scale L depend sig-

¹²that are relevant to lipid binding

¹³own estimate

¹⁴resulting in diffusion-limited binding; using the formulas from [134] and the values of the other parameters in our system, the diffusion-limited rate of binding to the cylinder is estimated to be $\sim 10^{-12} \frac{\text{m}^2}{\text{s}}$, while the estimate for the interparticle reaction on the membrane is $\sim 10^{-14} \frac{\text{m}^2}{\text{s}}$.

nificantly on j or k_u ; consequently, as seen in panel C, in contrast to the trans-model, the anticorrelation between A and L is virtually non-existent. This is not surprising, since in the cis-model the density-dependent amplification of local autophosphorylation that forms the basis of the buffering effect in the trans-model is not present. Rather, the subsequent autonomous phosphorylation steps of the Pom1 particles in the cis-model constitute a cascade of events with exponentially distributed times, at the end of which unbinding can occur; the total time until unbinding thus is a gamma-distributed random variable. The benefit of the cis-autophosphorylation therefore is diametrically different as the one of trans-phosphorylation: cis-autophosphorylation mainly acts as a precise “timer” for the unbinding events, which keeps the distances travelled by the particles sharply distributed and thus makes the resulting gradient shapes extraordinarily robust to variations of the other parameters. To end with, panel D of Fig. S10 shows that also the cis-model operates in the Poissonian noise regime; this is equally unsurprising, because all processes involved in the gradient formation are first-order and Poissonian.

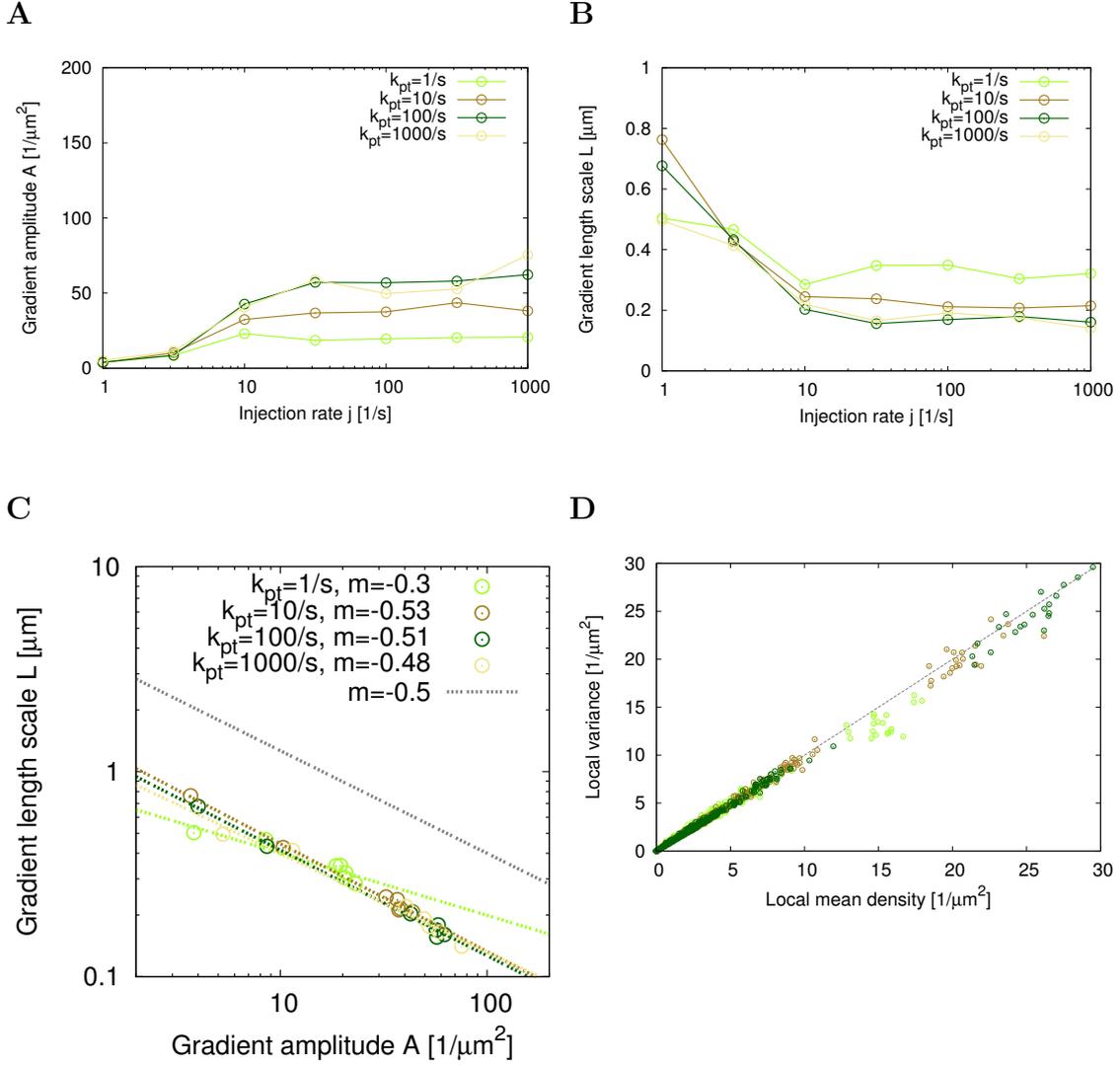


Figure S9: Gradient characteristics for different autophosphorylation rates (trans-model). Gradient amplitude (A) and length scale L (B), obtained by fitting an exponential function to stationary radial profiles, plotted against the injection rate j at the gradient origin for different in-complex phosphorylation-dissociation rates k_{pt} . Lines are guidelines for the eye. (C) Log-log plot of the gradient length L vs. gradient amplitude A for the same data. (D) Variance vs. mean for the local total membrane-bound Pom1 density, combining the values of all histogram bins into one scatter plot; point colors correspond to the colors in (A) - (C) (different values of k_{pt}).

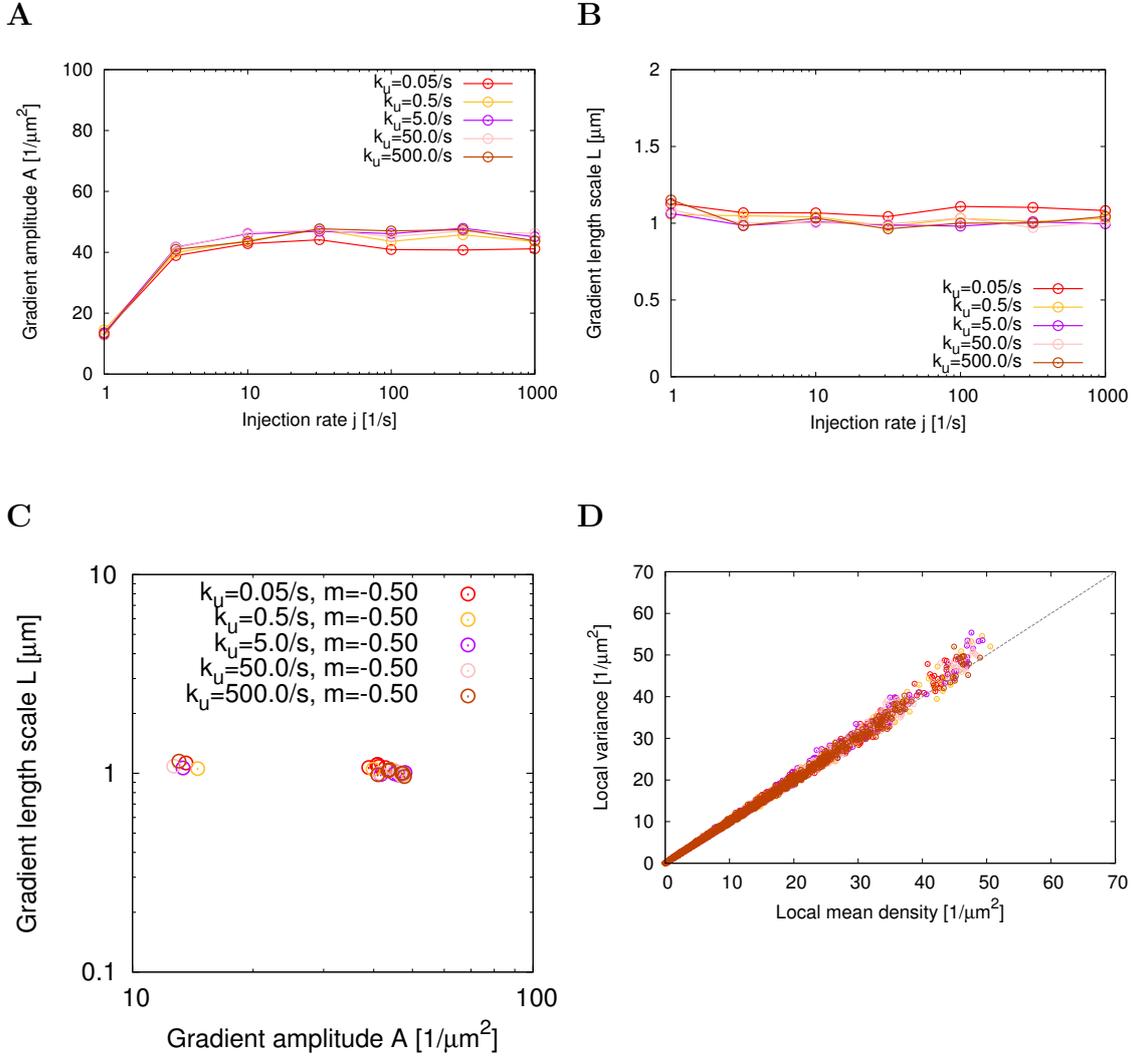


Figure S10: Gradient characteristics as a function of key parameters (cis-model). Gradient amplitude (A) and length scale L (B), obtained by fitting an exponential function to stationary radial profiles, plotted against the injection rate j at the gradient origin for different unbinding rates k_u of the fully phosphorylated Pom1. Lines are guidelines for the eye. (C) Log-log plot of the gradient length L vs. gradient amplitude A for the same data. (D) Variance vs. mean for the local total membrane-bound Pom1 density, combining the values of all histogram bins into one scatter plot; point colors correspond to the colors in (A) - (C) (different values of k_u).

References

- [1] Yu J, Xiao J, Ren X, Lao K, Xie XS (2006) Probing Gene Expression in Live Cells, One Protein Molecule at a Time. *Science* **311**:1600.
- [2] Elf J, Li GW, Xie XS (2007) Probing Transcription Factor Dynamics at the Single-Molecule Level in a Living Cell. *Science* **316**:1191.
- [3] Simicevic J, Schmid AW, Gilardoni PA, Zoller B, Raghav SK, Krier I, Gubelmann C, Lisacek F, Naef F, Moniatte M, Deplancke B (2013) Absolute quantification of transcription factors during cellular differentiation using multiplexed targeted proteomics. *Nat Meth* **10**:570–576.
- [4] Li GW, Burkhardt D, Gross C, Weissman J (2014) Quantifying Absolute Protein Synthesis Rates Reveals Principles Underlying Allocation of Cellular Resources. *Cell* **157**:624–635.
- [5] Garza de Leon F, Sellars L, Stracy M, Busby SJW, Kapanidis AN (2017) Tracking Low-Copy Transcription Factors in Living Bacteria: The Case of the lac Repressor. *Biophys J* **112**:1316–1327.
- [6] Zimmerman SB, Minton AP (1993) Macromolecular crowding: biochemical, biophysical, and physiological consequences. *Annu Rev Biophys Biomol Struct* **22**:27–65.
- [7] Ellis RJ (2001) Macromolecular crowding: an important but neglected aspect of the intracellular environment. *Curr Opin Struct Biol* **11**:114–119.
- [8] Ellis RJ (2001) Macromolecular crowding: obvious but underappreciated. *Trends Biochem Sci* **26**:597–604.
- [9] Li GW, Berg OG, Elf J (2009) Effects of macromolecular crowding and DNA looping on gene regulation kinetics. *Nat Phys* **5**:294–297.
- [10] Zhou HX (2013) Influence of crowded cellular environments on protein folding, binding, and oligomerization: Biological consequences and potentials of atomistic modeling. *FEBS Letters* **587**:1053–1061.
- [11] Dhugosz M, Trylska J (2011) Diffusion in crowded biological environments: applications of Brownian dynamics. *BMC Biophys* **4**:3.
- [12] Mahmutovic A, Fange D, Berg OG, Elf J (2012) Lost in presumption: stochastic reactions in spatial models. *Nat Meth* **9**:1163–1166.
- [13] Tsimring LS (2014) Noise in biology. *Rep Prog Phys* **77**:026601.
- [14] Elf J, Ehrenberg M (2004) Spontaneous separation of bi-stable biochemical systems into spatial domains of opposite phases. *Syst Biol (Stevenage)* **1**:230–236.
- [15] Lawson MJ, Petzold L, Hellander A (2015) Accuracy of the Michaelis–Menten approximation when analysing effects of molecular noise. *J R Soc Interface* **12**:20150054.
- [16] van Albada SB, ten Wolde PR (2007) Enzyme Localization Can Drastically Affect Signal Amplification in Signal Transduction Pathways. *PLoS Comput Biol* **3**:e195.
- [17] Morelli MJ, ten Wolde PR (2008) Reaction Brownian dynamics and the effect of spatial fluctuations on the gain of a push-pull network. *J Chem Phys* **129**:054112 1–11.
- [18] Takahashi K, Tănase-Nicola S, ten Wolde PR (2010) Spatio-temporal correlations can drastically change the response of a MAPK pathway. *Proc Natl Acad Sci USA* **107**:2473–2478.

- [19] Dushek O, van der Merwe PA, Shahrezaei V (2011) Ultrasensitivity in Multisite Phosphorylation of Membrane-Anchored Proteins. *Biophys J* **100**:1189–1197.
- [20] Jilkine A, Angenent SB, Wu LF, Altschuler SJ (2011) A Density-Dependent Switch Drives Stochastic Clustering and Polarization of Signaling Molecules. *PLoS Comput Biol* **7**:e1002271.
- [21] Mugler A, Tostevin F, ten Wolde PR (2013) Spatial partitioning improves the reliability of biochemical signaling. *Proc Natl Acad Sci USA* **110**:5927–5932.
- [22] Wehrens M, ten Wolde PR, Mugler A (2014) Positive feedback can lead to dynamic nanometer-scale clustering on cell membranes. *J Chem Phys* **141**:205102.
- [23] Morelli M, Allen R, ten Wolde PR (2011) Effects of Macromolecular Crowding on Genetic Networks. *Biophys J* **101**:2882–2891.
- [24] Klann MT, Lapin A, Reuss M (2009) Stochastic Simulation of Signal Transduction: Impact of the Cellular Architecture on Diffusion. *Biophys J* **96**:5122–5129.
- [25] Klann MT, Lapin A, Reuss M (2011) Agent-based simulation of reactions in the crowded and structured intracellular environment: Influence of mobility and location of the reactants. *BMC Syst Biol* **5**:71.
- [26] ten Wolde PR, Mugler A (2014) Chapter Twelve - Importance of Crowding in Signaling, Genetic, and Metabolic Networks. In: Jeon RHaKW, editor, *Int Rev Cell Mol Biol*, Academic Press, volume 307 of *New Models of the Cell Nucleus: Crowding, Entropic Forces, Phase Separation, and Fractals*. pp. 419–442.
- [27] van Zon JS, Morelli MJ, Tănase-Nicola S, ten Wolde PR (2006) Diffusion of Transcription Factors Can Drastically Enhance the Noise in Gene Expression. *Biophys J* **91**:4350 - 4367.
- [28] Mugler A, Gotway Bailey A, Takahashi K, ten Wolde PR (2012) Membrane clustering and the role of rebinding in biochemical signaling. *Biophys J* **102**:1069–1078.
- [29] Kaizu K, de Ronde W, Paijmans J, Takahashi K, Tostevin F, ten Wolde PR (2014) The Berg-Purcell Limit Revisited. *Biophys J* **106**:976–985.
- [30] Wang YM, Austin RH, Cox EC (2006) Single Molecule Measurements of Repressor Protein 1D Diffusion on DNA. *Phys Rev Lett* **97**:048302.
- [31] Bonnet I, Biebricher A, Pierre-Louis P, Loverdo C, Bénichou O, Voituriez R, Escudé C, Wende W, Pingoud A, Desbiolles P (2008) Sliding and jumping of single EcoRV restriction enzymes on non-cognate DNA. *Nucl Acids Res* **36**:4118–4127.
- [32] Tafvizi A, Huang F, Fersht AR, Mirny LA, van Oijen AM (2011) A single-molecule characterization of p53 search on DNA. *Proc Natl Acad Sci U S A* **108**:563–568.
- [33] Hammar P, Leroy P, Mahmutovic A, Marklund EG, Berg OG, Elf J (2012) The lac Repressor Displays Facilitated Diffusion in Living Cells. *Science* **336**:1595–1598.
- [34] Nguyen B, Sokoloski J, Galletto R, Elson EL, Wold MS, Lohman TM (2014) Diffusion of Human Replication Protein A along Single-Stranded DNA. *J Mol Biol* **426**:3246–3261.
- [35] Loverdo C, Bénichou O, Moreau M, Voituriez R (2008) Enhanced reaction kinetics in biological cells. *Nat Phys* **4**:134–137.
- [36] Loverdo C, Bénichou O, Voituriez R, Biebricher A, Bonnet I, Desbiolles P (2009) Quantifying Hopping and Jumping in Facilitated Diffusion of DNA-Binding Proteins. *Phys Rev Lett* **102**:188101.

- [37] Loverdo C, Bénichou O, Moreau M, Voituriez R (2009) Robustness of optimal intermittent search strategies in one, two, and three dimensions. *Phys Rev E* **80**:031146.
- [38] Bénichou O, Chevalier C, Klafter J, Meyer B, Voituriez R (2010) Geometry-controlled kinetics. *Nat Chem* **2**:472–477.
- [39] Bénichou O, Loverdo C, Moreau M, Voituriez R (2011) Intermittent search strategies. *Rev Mod Phys* **83**:81–129.
- [40] Paijmans J, ten Wolde PR (2014) Lower bound on the precision of transcriptional regulation and why facilitated diffusion can reduce noise in gene expression. *Phys Rev E* **90**:032708.
- [41] Schwarz K, Schröder Y, Qu B, Hoth M, Rieger H (2016) Optimality of Spatially Inhomogeneous Search Strategies. *Phys Rev Lett* **117**:068101.
- [42] Fange D, Elf J (2006) Noise-Induced Min Phenotypes in *E. coli*. *PLoS Comput Biol* **2**:e80.
- [43] Mugler A, ten Wolde PR (2013) The Macroscopic Effects of Microscopic Heterogeneity in Cell Signaling. In: Rice SA, Dinner AR, editors, *Adv Chem Phys*, John Wiley & Sons, Inc. pp. 373–396.
- [44] Dobrzyński M, Rodríguez JV, Kaandorp JA, Blom JG (2007) Computational methods for diffusion-influenced biochemical reactions. *Bioinformatics* **23**:1969–1977.
- [45] Sokolowski TR, ten Wolde PR (2017). Spatial-Stochastic Simulation of Reaction-Diffusion Systems. arXiv:1705.08669 [q-bio.MN].
- [46] Hattne J, Fange D, Elf J (2005) Stochastic reaction-diffusion simulation with MesoRD. *Bioinformatics* **21**:2923–2924.
- [47] Wang S, Elf J, Hellander S, Lötstedt P (2013) Stochastic Reaction-Diffusion Processes with Embedded Lower-Dimensional Structures. *Bull Math Biol* **76**:819–853.
- [48] Drawert B, Engblom S, Hellander A (2012) URDME: a modular framework for stochastic simulation of reaction-transport processes in complex geometries. *BMC Syst Biol* **6**:76.
- [49] Drawert B, Lawson MJ, Petzold L, Khammash M (2010) The diffusive finite state projection algorithm for efficient simulation of the stochastic reaction-diffusion master equation. *J Chem Phys* **132**:074101.
- [50] Lampoudi S, Gillespie DT, Petzold LR (2009) The multinomial simulation algorithm for discrete stochastic simulation of reaction-diffusion systems. *J Chem Phys* **130**:094104.
- [51] Isaacson S, Peskin C (2006) Incorporating Diffusion in Complex Geometries into Stochastic Chemical Kinetics Simulations. *SIAM J Sci Comput* **28**:47–74.
- [52] Drawert B, Hellander A, Bales B, Banerjee D, Bellesia G, Daigle BJ Jr, Douglas G, Gu M, Gupta A, Hellander S, Horuk C, Nath D, Takkar A, Wu S, Lötstedt P, Krintz C, Petzold LR (2016) Stochastic Simulation Service: Bridging the Gap between the Computational Expert and the Biologist. *PLoS Comput Biol* **12**:1-15.
- [53] Moraru II, Schaff JC, Slepchenko BM, Blinov ML, Morgan F, Lakshminarayana A, Gao F, Li Y, Loew LM (2008) Virtual Cell modelling and simulation software environment. *IET Syst Biol* **2**:352–362.

- [54] Rodríguez JV, Kaandorp JA, Dobrzyński M, Blom JG (2006) Spatial stochastic modelling of the phosphoenolpyruvate-dependent phosphotransferase (PTS) pathway in *Escherichia coli*. *Bioinformatics* **22**:1895–1901.
- [55] Vigelius M, Lane A, Meyer B (2010) Accelerating Reaction-Diffusion Simulations with General-Purpose Graphics Processing Units. *Bioinformatics* :btq622.
- [56] Gillespie D (1976) A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J Comput Phys* **22**:403–434.
- [57] Gillespie D (1977) Exact stochastic simulation of coupled chemical reactions. *J Chem Phys* **81**:2340–2361.
- [58] Andrews SS, Bray D (2004) Stochastic simulation of chemical reactions with spatial resolution and single molecule detail. *Phys Biol* **1**:137.
- [59] Andrews SS, Addy NJ, Brent R, Arkin AP (2010) Detailed Simulations of Cell Biology with Smoldyn 2.1. *PLoS Comput Biol* **6**:e1000705.
- [60] Stiles JR, van Helden D, Bartol TM, Salpeter EE, Salpeter MM (1996) Miniature endplate current rise times <100 microseconds from improved recordings can be modeled with passive acetylcholine diffusion from a synaptic vesicle. *Proc Natl Acad Sci USA* **93**:5747–5752.
- [61] Stiles JR, Bartol TM (2000) Monte Carlo Methods for Simulating Realistic Synaptic Microphysiology Using MCell. In: Computational Neuroscience, CRC Press, Frontiers in Neuroscience.
- [62] Franks KM, Bartol TM, Sejnowski TJ (2002) A Monte Carlo Model Reveals Independent Signaling at Central Glutamatergic Synapse neuromuscular junction. *Biophys J* **83**:2333–2348.
- [63] Kerr R, Bartol T, Kaminsky B, Dittrich M, Chang J, Baden S, Sejnowski T, Stiles J (2008) Fast Monte Carlo Simulation Methods for Biological Reaction-Diffusion Systems in Solution and on Surfaces. *SIAM J Sci Comput* **30**:3126–3149.
- [64] Stefan MI, Bartol TM, Sejnowski TJ, Kennedy MB (2014) Multi-state Modeling of Biomolecules. *PLoS Comput Biol* **10**:e1003844.
- [65] Plimpton SJ, Slepoy A (2005) Microbial cell modeling via reacting diffusing particles. *J Phys Conf Ser* **16**:305–309.
- [66] Boulianne L, Assaad SA, Dumontier M, Gross WJ (2008) GridCell: a stochastic particle-based biological system simulator. *BMC Syst Biol* **2**:66.
- [67] Arjunan SNV, Tomita M (2009) A new multicompartmental reaction-diffusion modeling method links transient membrane attachment of *E. coli*. *Syst Synth Biol* **4**:35–53.
- [68] Miyauchi A, Iwamoto K, Arjunan SNV, Takahashi K (2016) pSpatocyte: A Parallel Stochastic Method for Particle Reaction-Diffusion Systems. *arXiv:160503726 [q-bio]* .
- [69] Schöneberg J, Noé F (2013) ReaDDy - A Software for Particle-Based Reaction-Diffusion Dynamics in Crowded Cellular Environments. *PLoS ONE* **8**:e74261.
- [70] van Zon JS, ten Wolde PR (2005) Simulating biochemical networks at the particle level and in time and space: Green’s function reaction dynamics. *Phys Rev Lett* **94**:128103 1–4.
- [71] van Zon JS, ten Wolde PR (2005) Green’s-function reaction dynamics: a particle-based approach for simulating biochemical networks in time and space. *J Chem Phys* **123**:234910 1–16.

- [72] Janmey PA (1998) The cytoskeleton and cell signaling: component localization and mechanical coupling. *Physiol Rev* **78**:763–781.
- [73] Kholodenko BN (2006) Cell-signalling dynamics in time and space. *Nat Rev Mol Cell Biol* **7**:165–176.
- [74] Moseley JB, Goode BL (2006) The Yeast Actin Cytoskeleton: from Cellular Function to Biochemical Mechanism. *Microbiol Mol Biol Rev* **70**:605–645.
- [75] Kuchler A, Yoshimoto M, Luginbuhl S, Mavelli F, Walde P (2016) Enzymatic reactions in confined environments. *Nat Nano* **11**:409–420.
- [76] Ooppelstrup T, Bulatov VV, Gilmer GH, Kalos MH, Sadigh B (2006) First-Passage Monte Carlo Algorithm: Diffusion without All the Hops. *Phys Rev Lett* **97**:230602.
- [77] Ooppelstrup T, Bulatov VV, Donev A, Kalos MH, Gilmer GH, Sadigh B (2009) First-passage kinetic Monte Carlo method. *Phys Rev E* **80**:066701.
- [78] Donev A, Bulatov VV, Ooppelstrup T, Gilmer GH, Sadigh B, Kalos MH (2010) A first-passage kinetic Monte Carlo algorithm for complex reaction–diffusion systems. *J Comp Phys* **229**:3214–3236.
- [79] Hellander S, Lötstedt P (2011) Flexible single molecule simulation of reaction–diffusion processes. *J Comp Phys* **230**:3948–3965.
- [80] Schwarz K, Rieger H (2013) Efficient kinetic Monte Carlo method for reaction–diffusion problems with spatially varying annihilation rates. *J Comp Phys* **237**:396–410.
- [81] Smoluchowski M (1915) Über Brownsche Molekularbewegung unter Einwirkung äußerer Kräfte und den Zusammenhang mit der verallgemeinerten Diffusionsgleichung. *Ann Phys* **353**:1103–1112.
- [82] Ross JL, Ali MY, Warshaw DM (2008) Cargo Transport: Molecular Motors Navigate a Complex Cytoskeleton. *Curr Opin Cell Biol* **20**:41–47.
- [83] Hirokawa N, Noda Y, Tanaka Y, Niwa S (2009) Kinesin superfamily motor proteins and intracellular transport. *Nat Rev Mol Cell Biol* **10**:682–696.
- [84] Verhey KJ, Hammond JW (2009) Traffic control: regulation of kinesin motors. *Nat Rev Mol Cell Biol* **10**:765–777.
- [85] Tischer C, ten Wolde PR, Dogterom M (2010) Providing positional information with active transport on dynamic microtubules. *Biophys J* **99**:726–735.
- [86] Akhmanova A, Dogterom M (2011) Kinesins Lead Aging Microtubules to Catastrophe. *Cell* **147**:966–968.
- [87] Hammer III JA, Sellers JR (2012) Walking to work: roles for class V myosins as cargo transporters. *Nat Rev Mol Cell Biol* **13**:13–26.
- [88] Hancock WO (2014) Bidirectional cargo transport: moving beyond tug of war. *Nat Rev Mol Cell Biol* **15**:615–628.
- [89] Khaitlina SY (2014) Intracellular transport based on actin polymerization. *Biochemistry Mosc* **79**:917–927.
- [90] Huber F, Boire A, López MP, Koenderink GH (2015) Cytoskeletal crosstalk: when three different personalities team up. *Current Opinion in Cell Biology* **32**:39–47.
- [91] Morelli MJ, ten Wolde PR (2008) Reaction Brownian dynamics and the effect of spatial fluctuations on the gain of a push-pull network. *J Chem Phys* **129**:054112 1–11.

- [92] Bastiaens P, Caudron M, Niethammer P, Karsenti E (2006) Gradients in the self-organization of the mitotic spindle. *Trends in Cell Biol* **16**:125–134.
- [93] Niethammer P, Bastiaens P, Karsenti E (2004) Stathmin-Tubulin Interaction Gradients in Motile and Mitotic Cells. *Science* **303**:1862–1866.
- [94] Siegrist SE, Doe CQ (2007) Microtubule-induced cortical cell polarity. *Genes & Development* **2007**:483–496.
- [95] Martin SG (2009) Microtubule-dependent cell morphogenesis in the fission yeast. *Trends Cell Biol* **19**:447–454.
- [96] Lo Presti L, Martin SG (2011) Shaping fission yeast cells by rerouting actin-based transport on microtubules. *Curr Biol* **21**:2064–2069.
- [97] Howard M (2012) How to build a robust intracellular concentration gradient. *Trends in Cell Biol* **22**:311–317.
- [98] Kruse K (2012) Bacterial Organization in Space and Time. In: Comprehensive Biophysics, Oxford: Academic Press, volume 7. pp. 208–221.
- [99] Schmick M, Bastiaens PIH (2014) The interdependence of membrane shape and cellular signal processing. *Cell* **156**:1132–1138.
- [100] Recouvreux P, Sokolowski TR, Gramoustianou A, ten Wolde PR, Dogterom M (2016) Chimera proteins with affinity for membranes and microtubule tips polarize in the membrane of fission yeast cells. *Proc Natl Acad Sci USA* **113**:1811–1816.
- [101] Bähler J, Pringle JR (1998) Pom1p, a fission yeast protein kinase that provides positional information for both polarized growth and cytokinesis. *Genes & Development* **12**:1356–1370.
- [102] Moseley JB, Mayeux A, Paoletti A, Nurse P (2009) A spatial gradient coordinates cell size and mitotic entry in fission yeast. *Nature* **459**:857–860.
- [103] Martin SG, Berthelot-Grosjean M (2009) Polar gradients of the DYRK-family kinase Pom1 couple cell length with the cell cycle. *Nature* **459**:852–856.
- [104] Tostevin F (2011) Precision of Sensing Cell Length via Concentration Gradients. *Biophys J* **100**:294–303.
- [105] Vilela M, Morgan JJ, Lindahl PA (2010) Mathematical Model of a Cell Size Checkpoint. *PLoS Comput Biol* **6**:e1001036.
- [106] Hachet O, Berthelot-Grosjean M, Kokkoris K, Vincenzetti V, Moosbrugger J, Martin SG (2011) A Phosphorylation Cycle Shapes Gradients of the DYRK Family Kinase Pom1 at the Plasma Membrane. *Cell* **145**:1116–1128.
- [107] Lochhead PA, Sibbet G, Morrice N, Cleghon V (2005) Activation-Loop Autophosphorylation Is Mediated by a Novel Transitional Intermediate Form of DYRKs. *Cell* **121**:925–936.
- [108] Lochhead PA (2009) Protein Kinase Activation Loop Autophosphorylation in Cis: Overcoming a Catch-22 Situation. *Sci Signal* **2**:pe4.
- [109] Saul VV, de la Vega L, Milanovic M, Krüger M, Braun T, Fritz-Wolf K, Becker K, Schmitz ML (2013) HIPK2 kinase activity depends on cis-autophosphorylation of its activation loop. *J Mol Cell Biol* **5**:27–38.
- [110] Hersch M, Hachet O, Dalessi S, Ullal P, Bhatia P, Bergmann S, Martin SG (2015) Pom1 gradient buffering through intermolecular auto-phosphorylation. *Mol Syst Biol* **11**:818.

- [111] Dematté L (2010) Parallel Particle-Based Reaction Diffusion: A GPU Implementation. In: 2010 Ninth International Workshop on Parallel and Distributed Methods in Verification, and Second International Workshop on High Performance Computational Systems Biology. pp. 67–77.
- [112] Gladkov DV, Alberts S, D’Souza RM, Andrews SS (2011) Accelerating the Smoldyn spatial stochastic biochemical reaction network simulator using GPUs. *Proceedings of the 19th High Performance Computing Symposia* :151–158.
- [113] Dematté L (2012) Smoldyn on graphics processing units: massively parallel Brownian dynamics simulations. *IEEE/ACM Trans Comput Biol Bioinform* **9**:655–667.
- [114] Erban R, Flegg MB, Papoian GA (2013) Multiscale Stochastic Reaction–Diffusion Modeling: Application to Actin Dynamics in Filopodia. *Bull Math Biol* **76**:799–818.
- [115] Franz B, Flegg M, Chapman S, Erban R (2013) Multiscale Reaction-Diffusion Algorithms: PDE-Assisted Brownian Dynamics. *SIAM J Appl Math* **73**:1224–1247.
- [116] Flegg M, Chapman S, Zheng L, Erban R (2014) Analysis of the Two-Regime Method on Square Meshes. *SIAM J Sci Comput* **36**:B561–B588.
- [117] Flegg MB, Hellander S, Erban R (2015) Convergence of methods for coupling of microscopic and mesoscopic reaction–diffusion simulations. *J Comput Phys* **289**:1–17.
- [118] Robinson M, Flegg M, Erban R (2014) Adaptive two-regime method: Application to front propagation. *J Chem Phys* **140**:124109.
- [119] Robinson M, Andrews SS, Erban R (2015) Multiscale reaction-diffusion simulations with Smoldyn. *Bioinformatics* :btv149.
- [120] Vijaykumar A, Bolhuis PG, ten Wolde PR (2015) Combining molecular dynamics with mesoscopic Green’s function reaction dynamics simulations. *J Chem Phys* **143**:214102.
- [121] Vijaykumar A, Ouldridge TE, ten Wolde PR, Bolhuis PG (2017) Multiscale simulations of anisotropic particles combining molecular dynamics and Green’s function reaction dynamics. *J Chem Phys* **146**:114106.
- [122] Kaizu K, Takahashi K (2017) In preparation, personal communication.
- [123] Paijmans J (2012) The fundamental lower bound of the noise in transcriptional regulation. Master’s thesis, Univeristy of Amsterdam.
- [124] Smoluchowski M (1916) Drei Vorträge über Diffusion, Brown’sche Molekularbewegung und Koagulation von Kolloidteilchen. *Physik Z* **17**:557–551 and 587–599.
- [125] Carslaw HS, Jaeger JC (1959) Conduction of Heat in Solids. Oxford University Press, 2nd edition.
- [126] Lamm G, Schulten K (1983) Extended Brownian dynamics. II. Reactive, nonlinear diffusion. *J Chem Phys* **78**:2713–2734.
- [127] Watson GN (1962) A treatise on the theory of Bessel functions. Cambridge University Press, 2nd edition.
- [128] Bossen L (2010) Integrating membranes into the enhanced Green’s Function Reaction Dynamics algorithm. Master’s thesis, University of Amsterdam.
- [129] Galjart N, Perez F (2003) A plus-end raft to control microtubule dynamics and function. *Curr Opin Cell Biol* **15**:48–53.

- [130] Sawin KE (**2000**) Microtubule dynamics: The view from the tip. *Curr Biol* **10**:R860–R862.
- [131] Halford SE, Marko JF (**2004**) How do site-specific DNA-binding proteins find their targets? *Nucleic Acids Res* **32**:3040–3052.
- [132] Shimamoto N (**1999**) One-dimensional Diffusion of Proteins along DNA. *J Biol Chem* **274**:15293–15296.
- [133] von Hippel PH, Berg OG (**1989**) Facilitated target location in biological systems. *J Biol Chem* **264**:675–678.
- [134] Kivenson A, Hagan MF (**2012**). Diffusion-limited rates on low-dimensional manifolds with extreme aspect ratios. arXiv:1208.5156 [q-bio.BM].
- [135] Saunders TE, Pan KZ, Angel A, Guan Y, Shah JV, Howard M, Chang F (**2012**) Noise reduction in the intracellular pom1p gradient by a dynamic clustering mechanism. *Dev Cell* **22**:558–572.