Kinetic Monte Carlo

Heiko Rieger

Theoretical Physics
Saarland University
Saarbrücken, Germany

DPG school on “Efficient Algorithms in Computational Physics”,
10.-14.9.2012, Bad Honnef

Intro

Kinetic Monte Carlo:

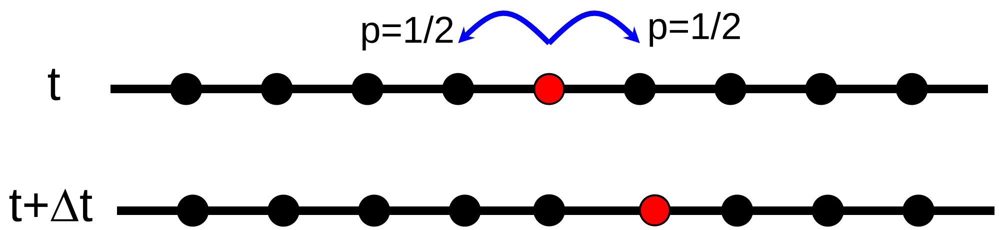
Simulation of the **dynamics** of stochastic processes

Simulation here means:

Generation / sampling of time sequences on a computer

Examples:

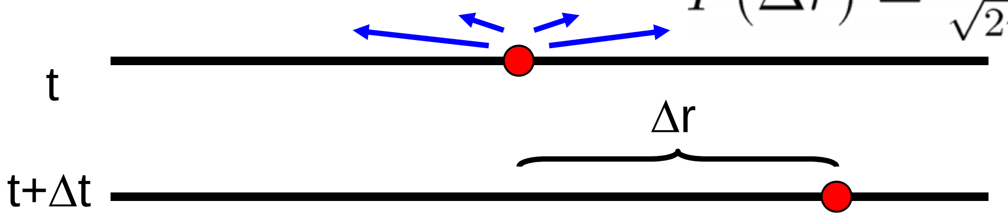
- Random walk in discrete time and discrete space



- Diffusion (Brownian dynamics) in discrete time and continuous space

$$P(\Delta r) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\Delta r^2/2\sigma^2}$$

$$\sigma = \sqrt{2D\Delta t}$$



Examples cont.

- Multi particle diffusion - with barriers (e.g. surface diffusion / epitaxial growth)
  - with collisions / exclusions (lattice gas, ASEP, …)
  - with chemical reactions (Reaction-diffusion simulations)

- Diffusion in continuous space AND continuous time
  - Greens function kinetic Monte Carlo
  - First passage time kinetic Monte Carlo

- **Chemical reactions**

  e.g. $\boxed{A + B \xrightarrow{k_1} C}$ (1) with (forward) reaction rate k_1

  System state: $S = (\#A, \#B, \#C)$

  Possible transitions for reaction 1: $S \rightarrow S' = (\#A-1, \#B-1, \#C+1)$

  Transition probability: $P(S',t+dt \mid S,t) = a_1 \cdot dt + O(dt^2)$

  Propensity $a_1 = k_1 \cdot \#A \cdot \#B$ (because of $\#A \cdot \#B$ possibilities for reaction 1)

  Transition rate $w(S \rightarrow S') = P(S',t+dt \mid S,t) / dt$

  (independent of time since process is Markov)

Chemical reactions cont.

Discrete state space (number of molecules of each species S=(#A,#B,#C,…)

Chemical reactions = stochastic process in this discrete state space

Example:

Reactions

$$A + B \xrightarrow{k_1} C$$

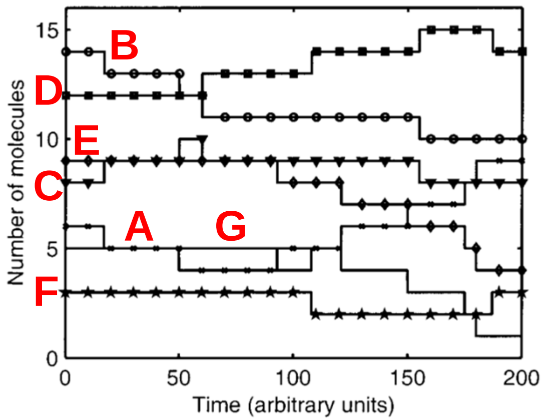
$$B + C \xrightarrow{k_2} D$$

$$D + E \xrightarrow{k_3} E + F$$

$$F \xrightarrow{k_4} D + G$$

$$E + G \xrightarrow{k_5} A$$

one process realization



Reaction times and types

Time	0	17	50	60	93	108	121	150	155	175
#A	6	5	4	4	5	5	6	7	7	8
#B	14	13	12	11	11	11	11	11	10	10
#C	8	9	10	9	9	9	9	9	8	8
#D	12	12	12	13	13	14	14	14	15	15
#E	9	9	9	9	8	8	7	6	6	5
#F	3	3	3	3	3	2	2	2	2	2
#G	5	5	5	5	4	5	4	3	3	2
Reaction	—	1	1	2	5	4	5	5	2	5

Master equation

$$\frac{d}{dt}P(S,t) = \sum_{S'} \left[\, w(S' \to S)\, P(S',t) - w(S \to S')\, P(S,t) \,\right]$$

For $0 \leq \#A \leq 9$, $0 \leq \#B \leq 9$, $0 \leq \#C \leq 9$, … Master eq has 10^7 coupled ODEs

too complex to solve numerically

Exact stochastic simulation

Remark: For large number of molecules and well stirred condition (spatial homogeneity)

deterministic mean-field description good

ODEs for $\rho_A(t)$=<#A>(t), $\rho_B(t)$=<#B>(t), …

neglects number fluctuations!!

$$\begin{aligned}
\frac{d}{dt}\rho_A(t) &= -k_1\rho_A(t)\rho_B(t) + k_2\rho \ldots \\
\frac{d}{dt}\rho_B(t) &= -k_1\rho_A(t)\rho_B(t) + k_3\rho \ldots \\
\frac{d}{dt}\rho_C(t) &= +k_1\rho_A(t)\rho_B(t) + k_4\rho \ldots
\end{aligned}$$

For small number of reaction partners: exact stochastic simulation

One reaction $S \rightarrow S'$:

$P(S',t+dt|\,S,t) = a_1 \cdot dt \;\Rightarrow$ Prob. That the transition $S \rightarrow S'$ happens at time τ

$$\boxed{P(S',t+\tau|S,t) = a_1 \exp(-\tau a_1)}$$ (Poisson process)

Numerical generation of transition times:

Generate exponentially distributed random numbers

$X = \ln y / a_1$, y uniformly distributed over [0,1]

Many reactions – like

$$\begin{aligned}
&A + B \xrightarrow{k_1} C \\
&B + C \xrightarrow{k_2} D \\
&D + E \xrightarrow{k_3} E + F \\
&F \xrightarrow{k_4} D + G \\
&E + G \xrightarrow{k_5} A
\end{aligned}$$

$\rightarrow$ Gillespie's algorithm

Gillespie's direct method

Basic problem in a simulation of a stochastic process with many possible transitions

- Which reaction occurs next?
- When does it occur?

Gillespie's answer [J. Comp. Phys. 22, 403 (1976)]:

Probability density $P(\mu,\tau)$ that the next reaction is μ and it occurs at time τ:

$$P(\mu,\tau)\,d\tau = a_\mu \exp\left(-\tau \sum_j a_j\right)\,d\tau$$

Probability distribution for reactions:
integrating (*) over $\tau \Rightarrow$

$$\text{prob}\,(\text{reaction } \mu) = a_\mu / \sum_j a_j$$

Probability distribution for times:
summing (*) over $\mu \Rightarrow$

$$P(\tau)d\tau = \left(\sum_j a_j\right) \exp\left(-\tau \sum_j a_j\right) d\tau$$

Related idea in standard Monte Carlo for Ising spin systems:
Bortz, Kalos, Lebowitz [J. Comp. Physics 17, 10 (1975)]

Gillespies direct algorithm

Algorithm 1. *Exact Stochastic Simulation — Direct Method (Gillespie)*

1. *Initialize(i.e., set initial numbers of molecules, set $t \leftarrow 0$).*
2. *Calculate the propensity function, a_i, for all i.*
3. *Choose μ according to the distribution* $\mathrm{prob}\,(\text{reaction } \mu) = a_\mu / \sum_j a_j$
4. *Choose τ according to an exponential with parameter $\sum_j a_j$* i.e. $P(\tau) = (\sum_j a_j) \cdot \exp(-\tau \sum_j a_j)$
5. *Change the number of molecules to reflect execution of reaction μ. Set $t \leftarrow t + \tau$.*
6. *Go to Step 2.*

Gillespies First Reaction Method

Generate a putative time τ_i for each reaction to occur (if no other reaction before)

Then choose the reaction μ whose putative time is first and let τ be τ_μ

Algorithm 2. *(Exact Stochastic Simulation — First Reaction Method)*

1. *Initialize(i.e., set initial numbers of molecules, set $t \leftarrow 0$).*
2. *Calculate the propensity function, a_i, for all i.*
3. *For each i, generate a putative time, τ_i, according to an exponential distribution with parameter a_i.*
4. *Let μ be the reaction whose putative time, τ_μ, is least.*
5. *Let τ be τ_μ.*
6. *Change the number of molecules to reflect execution of reaction μ. Set $t \leftarrow t + \tau$.*
7. *Go to Step 2.*

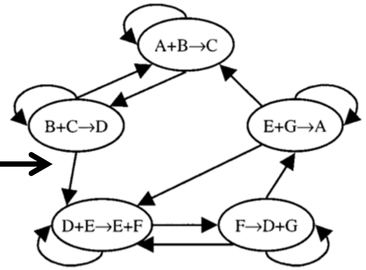
Computations in each iteration:

1) Update all r of the propensities a_i
2) Generate a putative time τ_i
3) Identify the smallest putative time τ_μ

Modification (next reaction method) will do away with each of these in turn

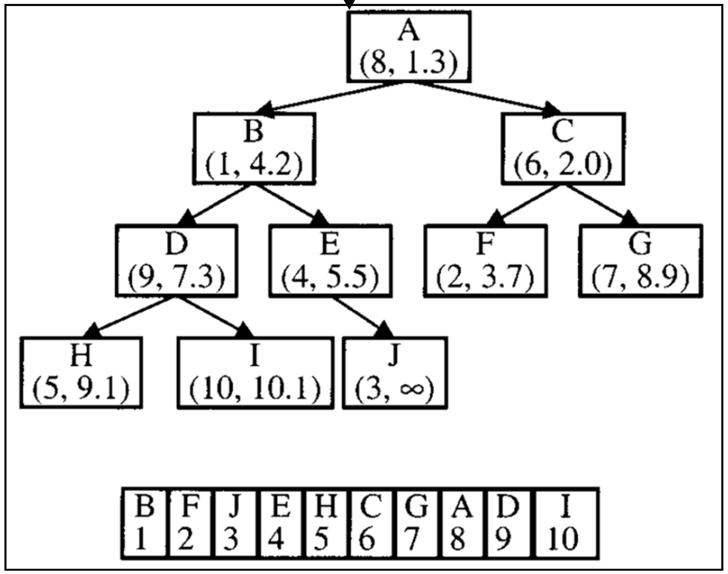
Next reaction method

- Store τ_i not just a_i
- Update only the minimum number of a_is (dependency graphs) →
- Re-use τ_is where appropriate
- Switch from relative times to absolute times
- Use appropriate data structures to store a_is and τ_is (indexed priority queue)



Algorithm 5. *(Exact Stochastic Simulation — Next Reaction Method)*

1. *Initialize:*
   (a) *set initial numbers of molecules, set $t \leftarrow 0$, generate a dependency graph $\mathcal{G}$;*
   (b) *calculate the propensity function, a_i, for all i;*
   (c) *for each i, generate a putative time, τ_i, according to an exponential distribution with parameter a_i;*
   (d) *store the τ_i values in an indexed priority queue $\mathcal{P}$.*
2. *Let μ be the reaction whose putative time, τ_μ, stored in $\mathcal{P}$, is least.*
3. *Let τ be τ_μ.*
4. *Change the number of molecules to reflect execution of reaction μ. Set $t \leftarrow \tau$.*
5. *For each edge (μ, α) in the dependency graph $\mathcal{G}$*
   (a) *update a_α;*
   (b) *if $\alpha \neq \mu$, set $\tau_\alpha \leftarrow (a_{\alpha,old}/a_{\alpha,new})(\tau_\alpha - t) + t$ (see note 11);*
   (c) *If $\alpha = \mu$, generate a random number, ρ, according to an exponential distribution with parameter a_μ, and set $\tau_\alpha \leftarrow \rho + t$;*
   (d) *replace the old τ_α value in $\mathcal{P}$ with the new value.*
6. *Go to Step 2.*



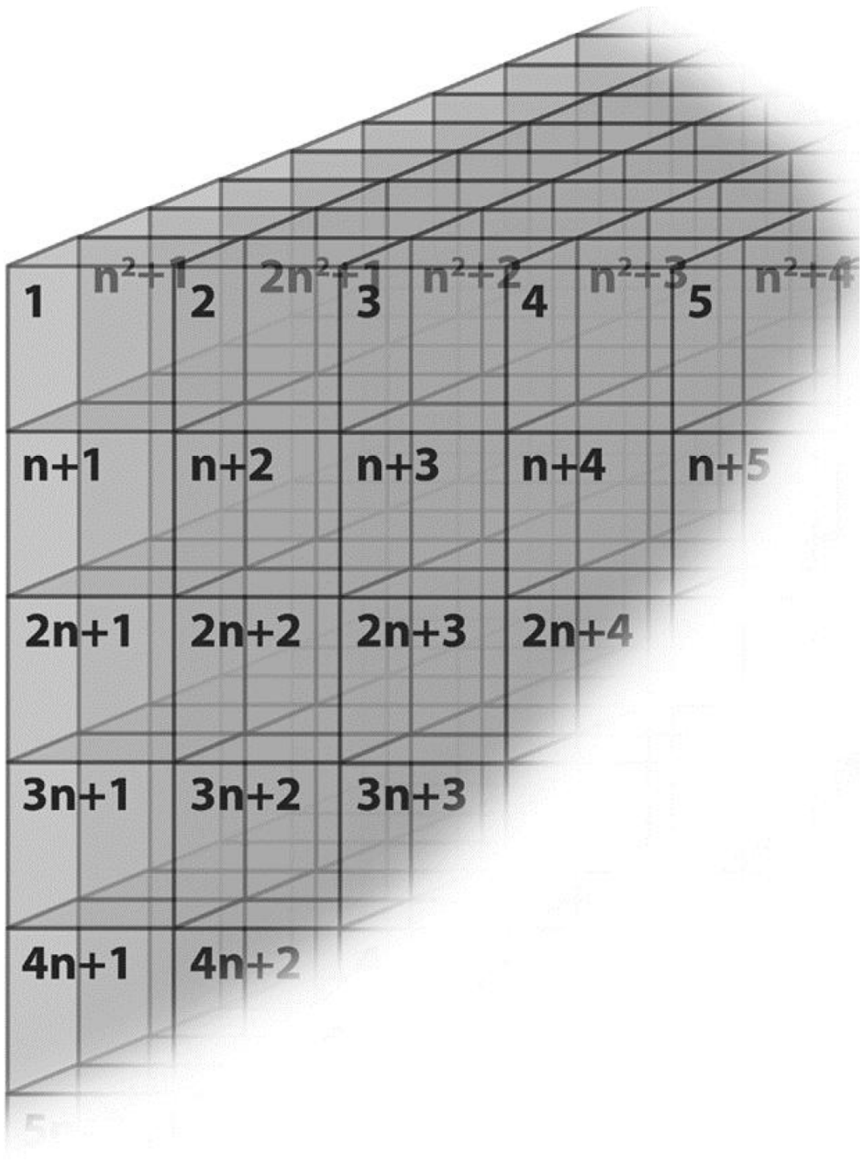
Event-driven algorithm, similar to MD-simulation of hard spheres or granular media

[Gibson, Bruck – J. Phys. Chem. A (2000)]

Reactions + Diffusion: Next sub-volume method

Not well stirred medium / non-uniform distribution of molecules:

Spatial inhomogeneity important!



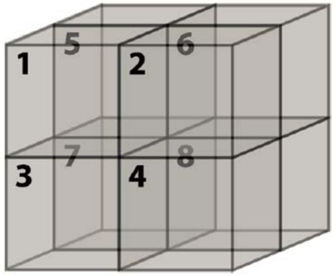
Particles diffuse in space:

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

Space discretization:
subvolumes of lateral size ℓ

$\Rightarrow$ random walk with jump rate

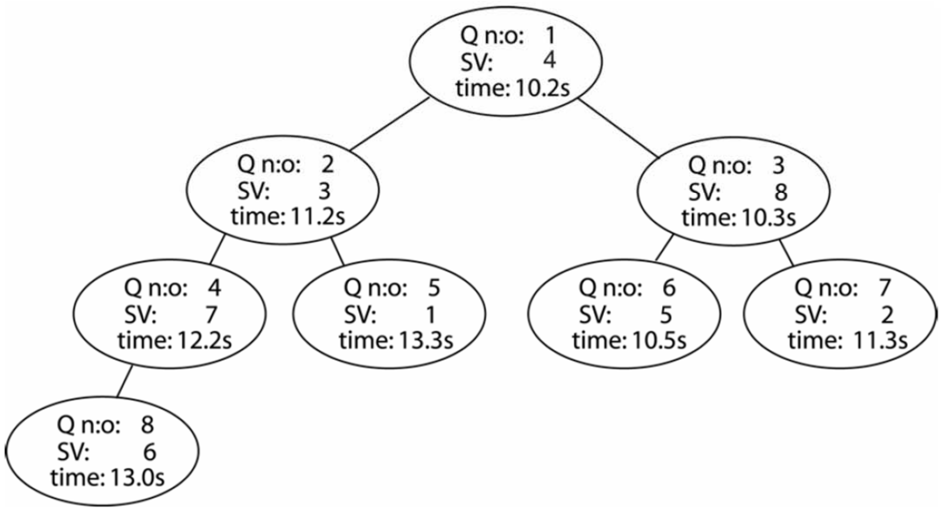
$$d^A_{\lambda\mu} = d^A_{\mu\lambda} = D/\ell^2$$



Next sub-volume method: Data structures

i	$n1$	$n2$	$n3$	$n4$	$n5$	$n6$	#A	#B	#C	r_i $[s^{-1}]$	s_i $[s^{-1}]$	r_i+s_i $[s^{-1}]$	Q
1	2	1	3	1	5	1	10	2	0	2.2	10	12.2	5
2	2	1	4	2	6	2	9	1	3	4.2	11.3	15.5	7
3	4	3	3	1	7	3	5	0	2	2.3	5.4	7.3	2
4	4	3	4	2	8	4	7	1	1	1.4	6.4	7.8	1
5	6	5	7	5	5	1	4	0	2	0.4	4.3	4.7	6
6	6	5	8	6	6	2	7	1	3	0.5	10.3	10.8	9
7	8	7	7	5	7	3	8	2	4	1.0	13.3	14.3	4
8	8	7	8	6	8	4	5	0	2	5.3	5.4	10.7	3

Connectivity matrix ($n1$–$n6$) | Configuration (#A, #B, #C) | Rate matrix (r_i, s_i, r_i+s_i) | Q-array (Q)



Event Queue

Position in Queue (Q)	Subvolume (SV)	τ_i (s)
1	4	10.2
2	3	11.2
3	8	10.3
4	7	12.2
5	1	13.3
6	5	10.5
7	2	11.3
8	6	13.0

Algorithm: Next sub-volume method

Initialization

1. Generate a *connectivity matrix* (see Fig.1, legend)
2. Distribute the initial numbers of molecules between the subvolumes and store them in the *configuration* matrix (see Fig.1, legend).
3. Calculate the sum, $r_\alpha = \sum_{j=1}^{R} a_{j\alpha}$, of intensities ($a_{j\alpha}$) for chemical reactions (j) in the subvolume α and store it in the *rate matrix* (Fig. 1, legend). The reaction intensities are calculated by using the volume Δ of the *SV* and the number of molecules in the *SV* to obtain the current concentrations.
4. Calculate the sum, $s_\alpha = \sum_{j=1}^{M} d_j X_j^\alpha$, of diffusion intensities $\left(d_j X_j^\alpha\right)$ in the subvolume α and store it in the *rate matrix*. The parameter $d_j = D_j/\ell^2$ is the rate constant for jumps between neighboring subvolumes for species j, as defined above. X_α^j is the number of molecules of species j in subvolume α and M is the number of different molecular species in the system.
5. Calculate the sum, $r_\alpha + s_\alpha$, for each subvolume and generate a random number, *rand*, uniformly distributed in [0,1]. This number samples the time for the first reaction-diffusion event in each subvolume as $t_\alpha = -\ln(rand)/(r_\alpha + s_\alpha)$.
6. Store the t_α in the *event queue array*, in such a way that all branches of the *event queue* are sorted with increasing event time. (Fig. 1, legend)

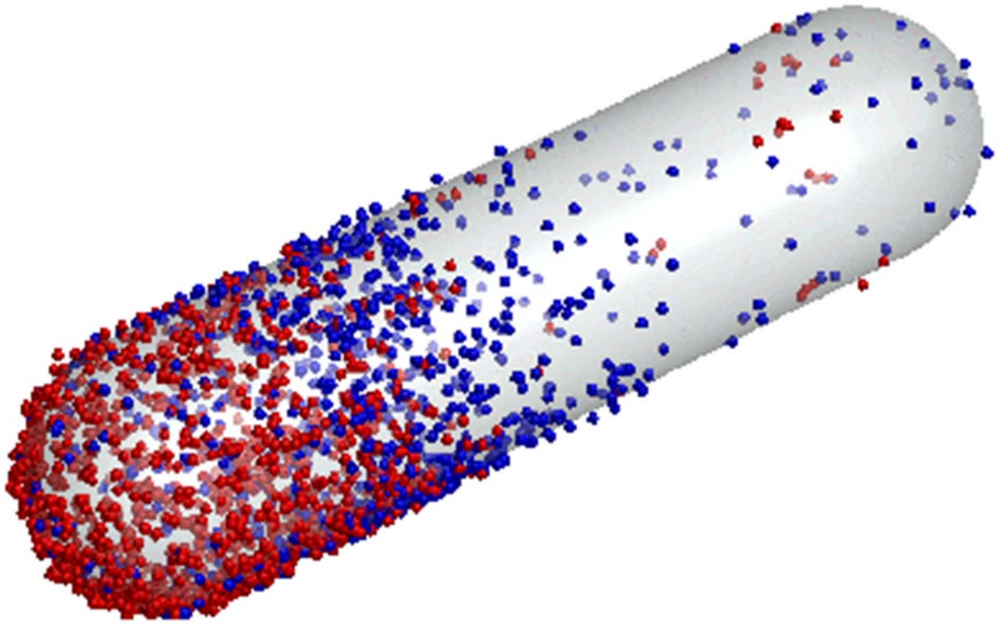
Iterations

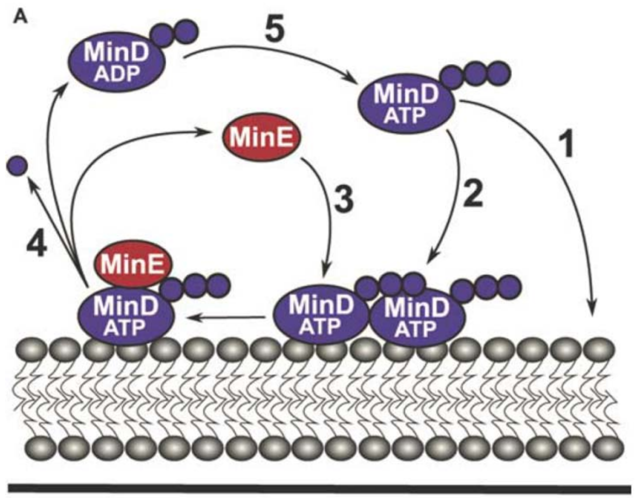
7. The next reaction-diffusion event will occur at time t_λ in the subvolume, $\alpha=\lambda$, that is at the top of the *event queue*. The event will be a chemical reaction if a newly generated $rand < r_\lambda/(r_\lambda + s_\lambda)$, and otherwise a jump out from the volume by diffusion.
8. Chemical reaction event ($rand < r_\lambda/(r_\lambda + s_\lambda)$)
   a. Rescale *rand* to [0,1], by dividing it with r_λ, and use the updated *rand* to sample which chemical reaction, i, that has occurred in subvolume λ according to the probability $P(i)=a_{i\lambda}/r_\lambda$.
   b. Update the elements in the configuration matrix that belong to the subvolume where the chemical event occurred.
   c. Recalculate the sum, $r_\lambda + s_\lambda$, in this subvolume and generate a new *rand* in [0,1] to obtain the time of the next reaction-diffusion event in this subvolume $t_\lambda^{next} = t_\lambda - \ln(rand)/(r_\lambda + s_\lambda)$.
   d. Reorder the branch of the event queue with subvolume λ according to the value of t_λ^{next} (see below).
9. Diffusion event ($rand > r_\lambda/(r_\lambda + s_\lambda)$).
   a. Rescale *rand* from paragraph 7. above according to ($rand - r_\lambda$)/(1- r_λ) and use the rescaled *rand* to sample which species, i, that diffused out from the subvolume according to the probability distribution $P(i) = d_i X_i^\lambda / s_\lambda$.
   b. The neighboring subvolume, γ, to which the diffusion event is targeted is sampled by randomly choosing one of the six columns in the connectivity matrix.
   c. Update the states of these *SVs* by removing a molecule of species i from *SV* λ and adding it to *SV* γ. Recalculate the sums, $r_\lambda + s_\lambda$ and $r_\gamma + s_\gamma$, for the SV and its neighbor where events have occurred. Generate two new random numbers, *rand*1 and *rand*2, and sample the times when the next reaction or diffusion events occurs in the subvolumes, $t_\lambda^{next} = t_\lambda - \ln(rand1)/(r_\lambda + s_\lambda)$ and $t_\gamma^{next} = t_\lambda - \ln(rand2)/(r_\gamma + s_\gamma)$.
   d. Reorder the *event queue* according to the values of t_λ^{next} and t_γ^{next} (see below).
10. Return to 7 for the next iteration.

Again: Event driven algorithm (now with diffusion events)!

[Elf, Ehrenberg – Syst. Biol. 2004]

Example: Min-system in Eschericha-Coli





B

(1) $MinD^{ATP}_{cyt} \xrightarrow{k_d} MinD_{mem}$ $\quad k_d = 0.0125 \mu m^{-1} s^{-1}$

(2) $MinD^{ATP}_{cyt} + MinD_{mem} \xrightarrow{k_{dD}} 2MinD_{mem}$ $\quad k_{dD} = 9 \cdot 10^6 M^{-1} s^{-1}$

(3) $MinE + MinD_{mem} \xrightarrow{k_{de}} MinDE$ $\quad k_{de} = 5.56 \cdot 10^7 M^{-1} s^{-1}$

(4) $MinDE \xrightarrow{k_e} MinD^{ADP}_{cyt} + MinE$ $\quad k_e = 0.7 s^{-1}$

(5) $MinD^{ADP}_{cyt} \xrightarrow{k^{ADP \to ATP}} MinD^{ATP}_{cyt}$ $\quad k^{ADP \to ATP} = 0.5 s^{-1}$

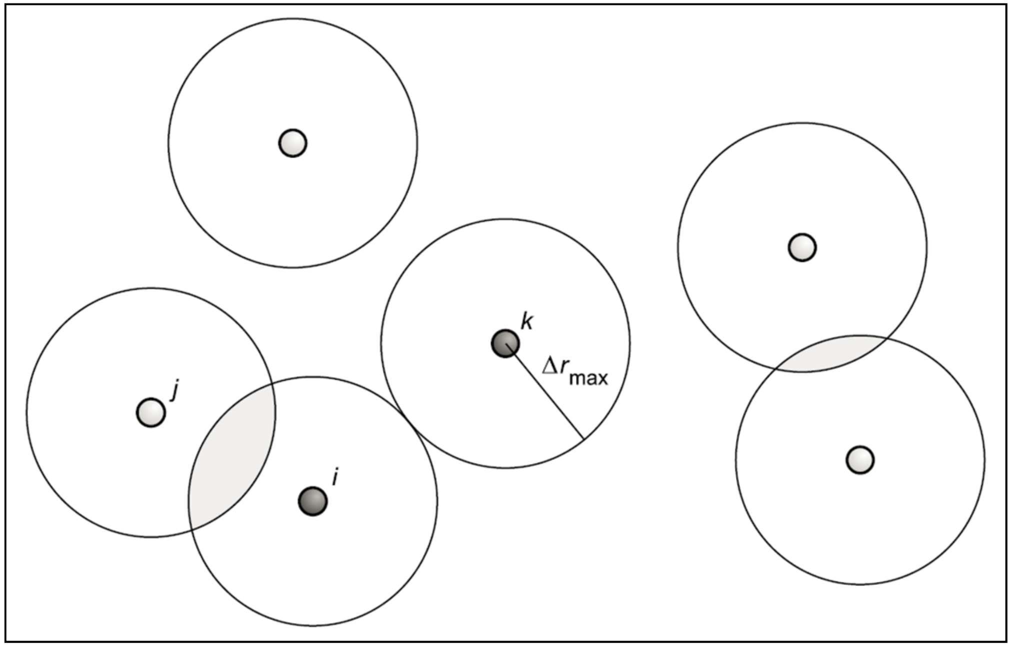
Sofware package for simulations of
Mesoscopic Reaction-Diffusion Systems – MESO-RD:
http://mesord.sourceforge.net

[Fange, Elf – PLoS 2006]

Reaction-diffusion systems in continuous space: Green's Function Reaction Dynamics

For low concentrations particles diffuse far before reacting:
$\Rightarrow$ Choose maximum time Δt_{max} such that
each particle can interact with at most one other particle within this time

$$\Delta r_{\max,i} = H\sqrt{6 D_i t_{\max,i}}$$



Green's function reaction dynamics

1 particle:

$$\partial_t p_1(\mathbf{r},t|\mathbf{r}_0,t_0) = D\nabla^2 p_1(\mathbf{r},t|\mathbf{r}_0,t_0).$$

$$p_1(\mathbf{r},t|\mathbf{r}_0,t_0) = \frac{1}{[4\pi D(t-t_0)]^{3/2}}\exp\left[-\frac{|\mathbf{r}-\mathbf{r}_0|^2}{4D(t-t_0)}\right]$$

2 particles (with interaction force **F**(**r**)):

$$\partial_t p_2(\mathbf{r}_A,\mathbf{r}_B,t|\mathbf{r}_{A0},\mathbf{r}_{B0},t_0) = [D_A\nabla_A^2 + D_B\nabla_B^2 - D_B\beta\nabla_B\cdot\mathbf{F}(\mathbf{r}) + D_A\beta\nabla_A\cdot\mathbf{F}(\mathbf{r})] \times p_2(\mathbf{r}_A,\mathbf{r}_B,t|\mathbf{r}_{A0},\mathbf{r}_{B0},t_0)$$

Separation in two independent processes:

$$\mathbf{R} = \sqrt{D_B/D_A}\,\mathbf{r}_A + \sqrt{D_A/D_B}\,\mathbf{r}_B,$$

$$\mathbf{r} = \mathbf{r}_B - \mathbf{r}_A,$$

interparticle distance

$$\boxed{\begin{aligned}\partial_t p_2^{\mathbf{R}}(\mathbf{R},t|\mathbf{R}_0,t_0) &= (D_A+D_B)\nabla_{\mathbf{R}}^2 \times p_2^{\mathbf{R}}(\mathbf{R},t|\mathbf{R}_0,t_0),\\ \partial_t p_2^{\mathbf{r}}(\mathbf{r},t|\mathbf{r}_0,t_0) &= (D_A+D_B)\nabla_{\mathbf{r}}\cdot(\nabla_{\mathbf{r}} - \mathbf{F}(\mathbf{r}))\\ &\quad\times p_2^{\mathbf{r}}(\mathbf{r},t_0|\mathbf{r},t_0),\ |\mathbf{r}| \geqslant \sigma.\end{aligned}}$$

Green's function reaction dynamics

Free diffusion of coordinate R:

$$p_2^{\mathbf{R}}(\mathbf{R},t|\mathbf{R}_0,t_0) = \frac{\exp[-|\mathbf{R}-\mathbf{R}_0|^2/4(D_A+D_B)(t-t_0)]}{[4\pi(D_A+D_B)(t-t_0)]^{3/2}}$$

Inter-particle coordinate r: Reaction (with rate k_a) taken into account as absorbing boundary condition at distance σ

$$p_2^{\mathbf{r}}(\mathbf{r},t_0|\mathbf{r}_0,t_0) = \delta(\mathbf{r}-\mathbf{r}_0),$$

$$p_2^{\mathbf{r}}(|\mathbf{r}| \to \infty,t|\mathbf{r}_0,t_0) = 0,$$

$$-j(\sigma,t|\mathbf{r}_0,t_0) \equiv 4\pi\sigma^2 D\left(\frac{\partial}{\partial r} - \mathbf{F}(\mathbf{r})\right)p_2^{\mathbf{r}}(\mathbf{r},t|\mathbf{r}_0,t_0)|_{|\mathbf{r}|=\sigma}$$

$$= k_a p_2^{\mathbf{r}}(|\mathbf{r}| = \sigma,t|\mathbf{r}_0,t_0),$$

j = outward radial flux of p_2 through contact surface area $4\pi\sigma^2$
(via reactions)

Core algorithm for GFRD:

For F=0: p_2 analytical solution,
For F≠0: numerical solution

Survival probability:

$$S_a(t|\mathbf{r}_0,t_0) = \int_{|\mathbf{r}|>\sigma} d\mathbf{r} p_2^{\mathbf{r}}(\mathbf{r},t|\mathbf{r}_0,t_0)$$

Probability per unit time that particle pair reacts at time:

$$q_a(t|\mathbf{r}_0,t_0) \equiv -\frac{\partial S_a(t|\mathbf{r}_0,t_0)}{\partial t}$$

Dissociation: $C \to A+B$:

Probability per unit time that next reaction occurs at time t:

$$q_d(t|t_0)dt = k_d \exp[-k_d(t-t_0)]dt$$

(1) If the system is in the dissociated state $A+B$, then draw a next association time t according to $q_a(t|\mathbf{r}_0,t_0)$

(a) If $(t-t_0) \geqslant \Delta t_{\max}$, then the two particles will not react within the time step; new positions for A and B at time $t_0+\Delta t_{\max}$ are obtained from $p_2^{\mathbf{R}}(\mathbf{R},t_0+\Delta t_{\max}|\mathbf{R}_0,t_0)$ and $p_2^{\mathbf{r}}(\mathbf{r},t_0+\Delta t_{\max}|\mathbf{r}_0,t_0)$

(b) If $(t-t_0) < \Delta t_{\max}$, then the next reaction will occur within the time step; a new position for particle C at time t is obtained from $p_2^{\mathbf{R}}(\mathbf{R},t|\mathbf{R}_0,t_0)$

(2) If the system is in the associated state C, then draw a next dissociation time from $q_d(t|t_0)$

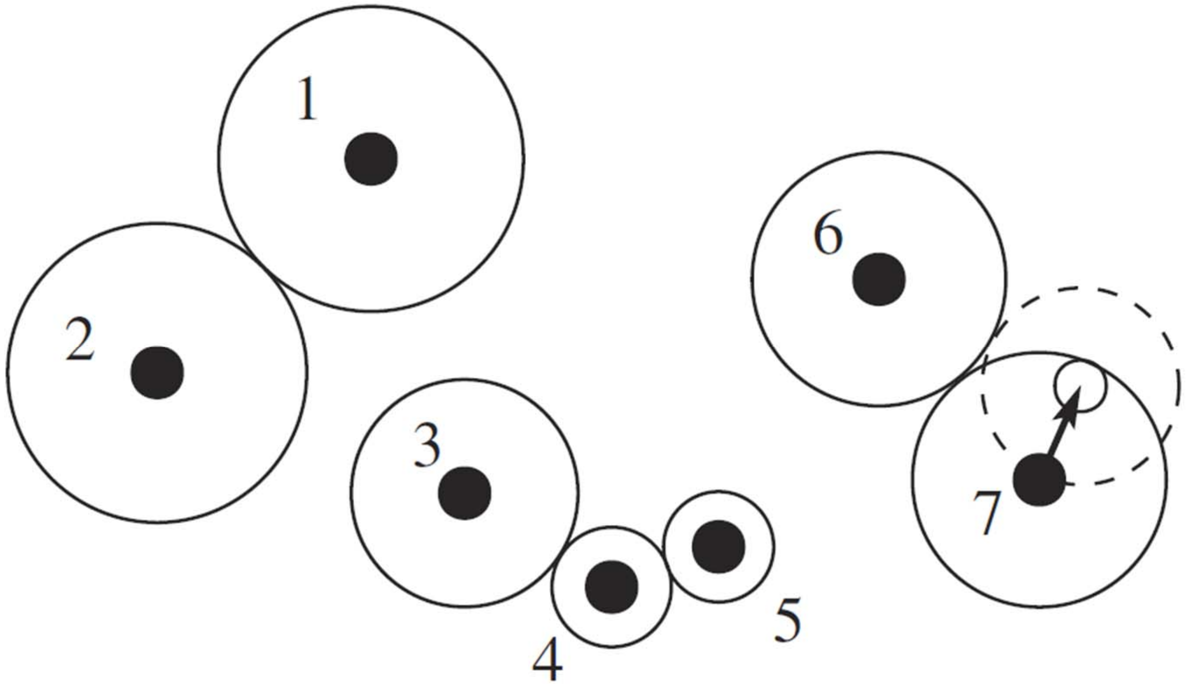
(a) If $(t-t_0) \geqslant \Delta t_{\max}$, then particle C will not have decayed by $t_0+\Delta t_{\max}$; a new position for particle C, $\mathbf{r}_C$, at time $t_0+\Delta t_{\max}$ is obtained from $p_1(\mathbf{r}_C,t_0+\Delta t_{\max}|\mathbf{r}_{C0},t_0)$

(b) If $(t-t_0) < \Delta t_{\max}$, the next reaction will occur within the maximum time step; the particles A and B are placed at time t adjacent to each other at positions around $\mathbf{r}_C$ as obtained from $p_1(\mathbf{r}_C,t|\mathbf{r}_{C0},t_0)$

[van Zon, ten Wolde – J. Chem. Phys. 2005]

First passage time kinetic Monte Carlo
„Diffusion without all the hops“

- protection zone (p.z.): domain around a particle with no other particles
- particles are freely diffusing within p.z.
- draw p.z. around each particle
- sample first passage time when a particle reaches the p.z. boundary
- propagate particle to boundary of p.z.
- update p.z.



[Oppelstrup et al – Phys. Rev. Lett. 2006]

First passage time kinetic Monte Carlo algorithm

(1) Set the global time clock to zero. Construct nonoverlapping protective domains around all walkers—use individual protection for single walkers and group protection for close pairs, as seems most efficient.

(2) Sample an exit time for each domain (in the case of protected pairs this can mean a scheduled collision). Put the sampled event times in an event queue (e.g., implemented as a heap), so that the shortest time can be efficiently found.

(3) Find the shortest exit time and identify the corresponding walker and domain. Sample the exit position for the selected walker. If the new position corresponds to a collision, take appropriate action.

(4) Check if any of the existing protective domains are close to the new position of the particle. If necessary to make more space available for protection of the propagated particle, use no-passage propagators to sample new locations for the particles in the neighboring domains.

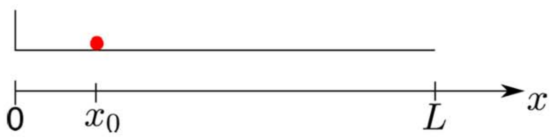
(5) Construct new protective domains for all particles that changed their positions in steps (3) or (4).

(6) Sample new event times for the particle(s) protected in step (5), as in step (2).

(7) Insert the new event time(s) into the event queue. Go to step (3).

Example: Sampling of first passage times in 1d

- New boundary conditions:
  - reflecting on the left: $\frac{\partial P(x,t|x_0,t_0)}{\partial x}|_{x=0} = 0$
  - absorbing on the right: $P(L,t|x_0,t_0) = 0$

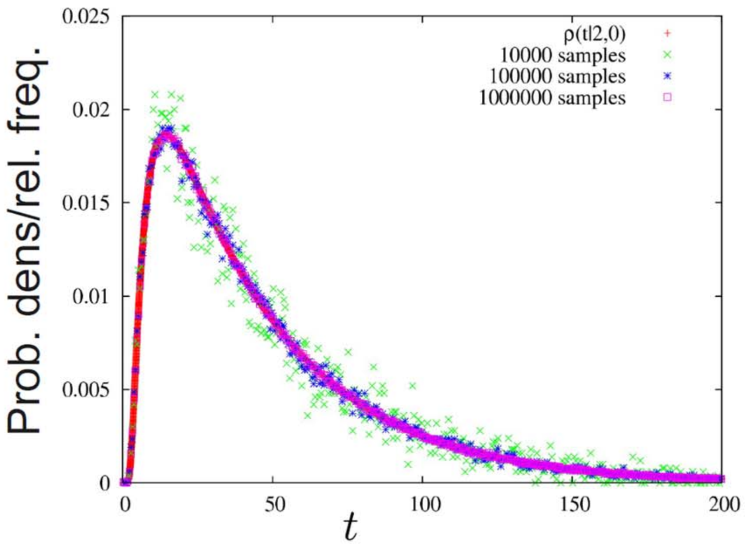


=> $P(x,t|x_0,t_0)$ is not normed for $t > t_0$.

- The probability for not having left the interval: $W(t|x_0,t_0) = \int_0^L \, dx \, P(x,t|x_0,t_0)$
- The probability density (in time) for leaving the interval: $$\rho(t|x_0,t_0) = -\frac{\partial W(t|x_0,t_0)}{\partial t}$$

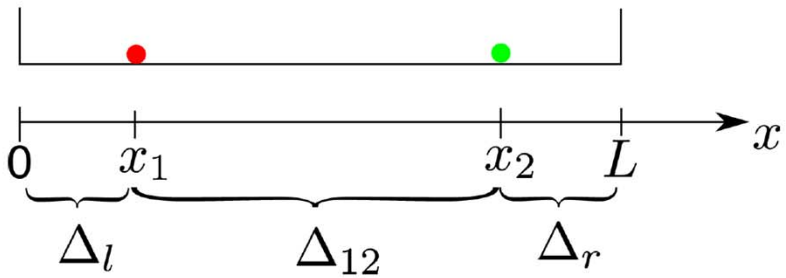
=> times for leaving the interval are sampled by applying the Inversion method on $1 - W$.

Example:



Histograms ($\Delta t = 0.5$) of sampled relative frequencies for leaving at time t, with $L = 10, \; x_0 = 2, \; D = 1$

Example: Two reacting particles in the interval

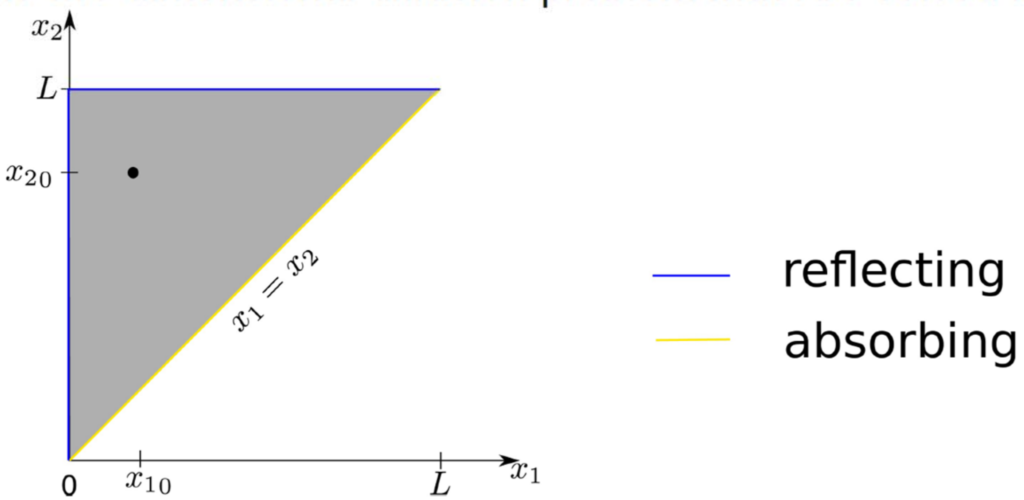


- If the particles vanish (reaction) at contact, the common probability density for the red and the green particle obeys:

$$\frac{\partial P(x_1,x_2,t|x_{10},x_{20},t_0)}{\partial t} = \left(D_1 \frac{\partial^2}{\partial {x_1}^2} + D_2 \frac{\partial^2}{\partial {x_2}^2} \right) P(x_1, x_2, t|x_{10}, x_{20}, t_0)$$

with the initial condition $P(x_1, x_2, t_0|x_{10}, x_{20}, t_0) = \delta(x_1 - x_{10}) \cdot \delta(x_2 - x_{20})$

This two dimensional diffusion problem must be solved in the gray triangle domain:



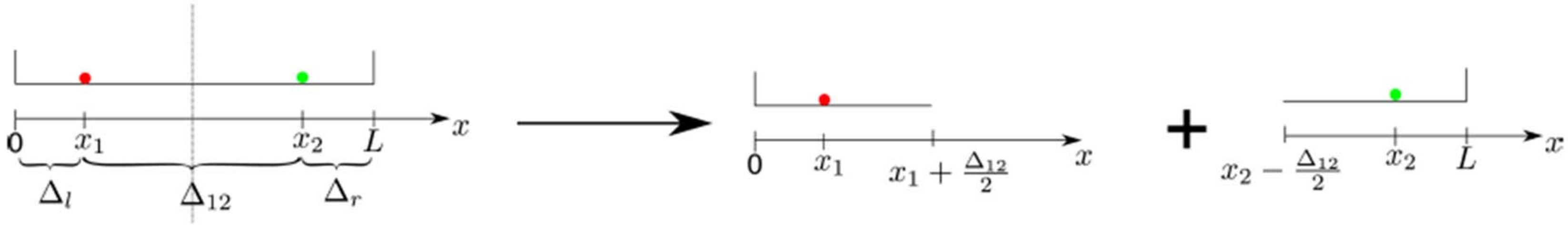
There is no analytic solution available for the probability density.

Strategy: Solve the problem in sequence of of easier problems.

Let's keep things simple $D_1 = D_2 = D$

- It is always possible to sample whether the green or the red particle reaches the middle position $1/2(x_2 - x_1)$ combined with a corresponding arrival time.

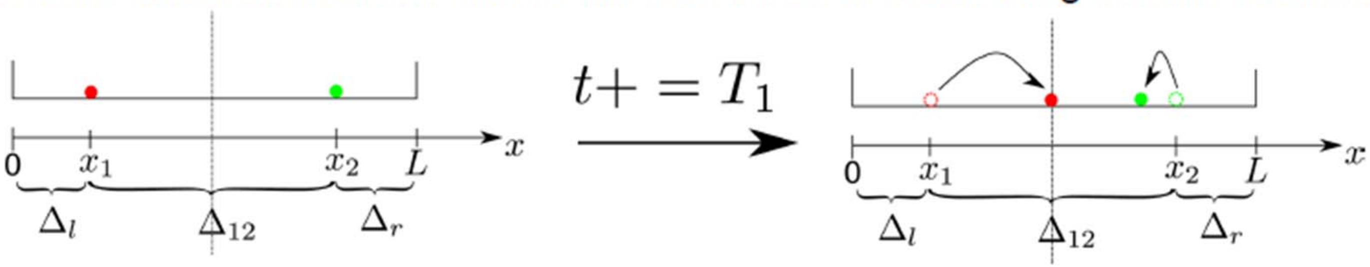
1) Split the system to two different first passage problems to the middle



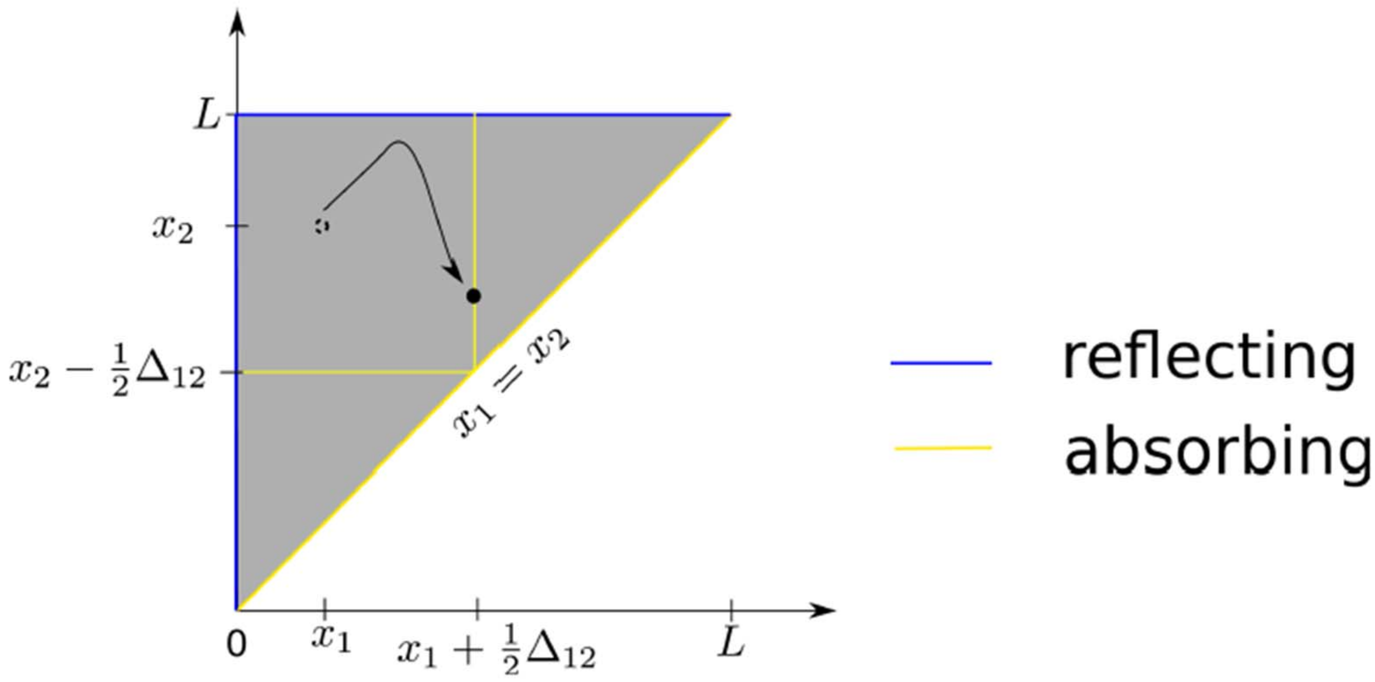
2) Solve them in the way it is shown before and look for the smallest arrival time.
For example, the arrival time T_1 of the red particle is smaller than the arrival time T_2 of the green one.

3) Update the system “Gillespie-like” :

-The time is incremented by T_1
-The red particle is moved to the middle
- For the green particle a new position in its subinterval is sampled with the Greens function method under the condition of not having left the subinterval.



In our 2d picture, this is equivalent to the first passage problem to the absorbing boundaries of an interior rectangle:



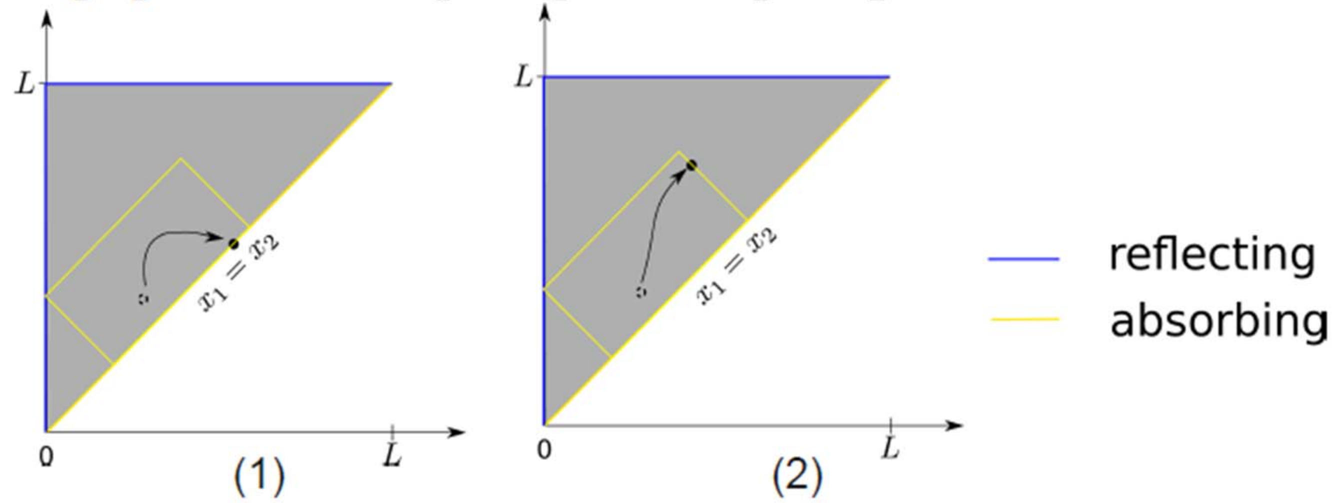
- This strategy can be continued arbitrary often, the particles will come arbitrary close to each other, but they will never meet.

  Two possibilites to overcome this problem:

  a) Stop, if a very small threshold-value is reached

     Apart from the fact, that it is “only” an approximation, there is a second disadvantage. If the particle-particle-distance is very small, the time incrementations per step will also become very small.

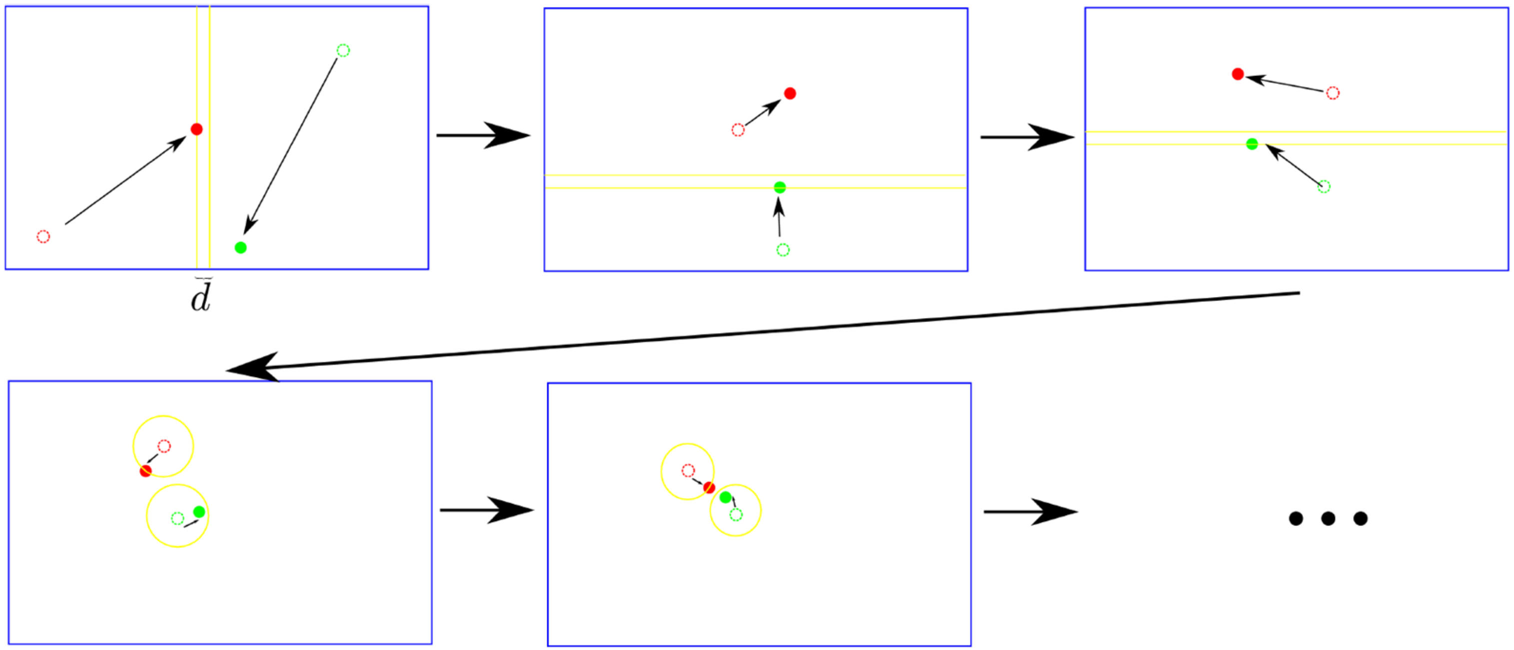
  b) Changing basis: $u = x_1 + x_2 \quad v = x_1 - x_2$



If situation (1) happens, the particles meet and the algorithm stops.
In the case of situation (2), we go on with a new step, either in the old or the new basis (depending on the ration between particle distance and the minimum particle-wall distance).

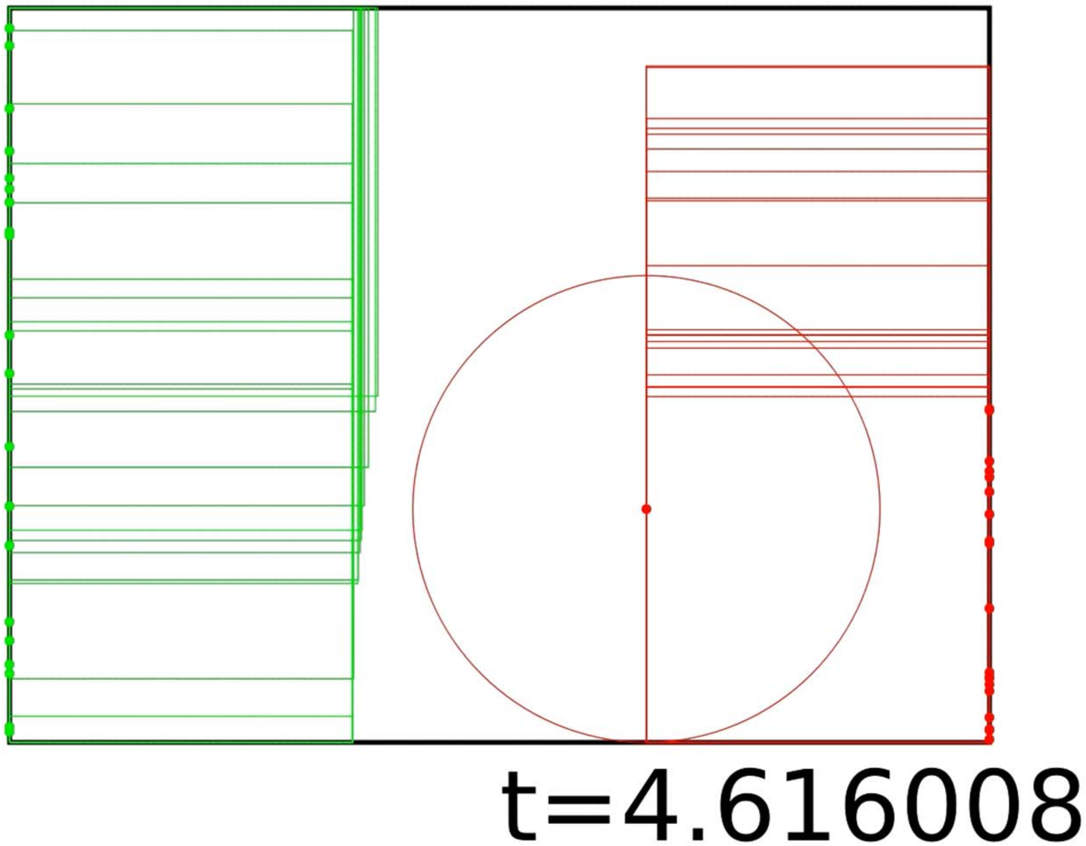
Example 2d: Two reacting particles in a box

- If the dimension is higher than 1, a minimum distance d (the sum of the particle radii) is needed
- The same strategy as before can also be used in a higher dimension:

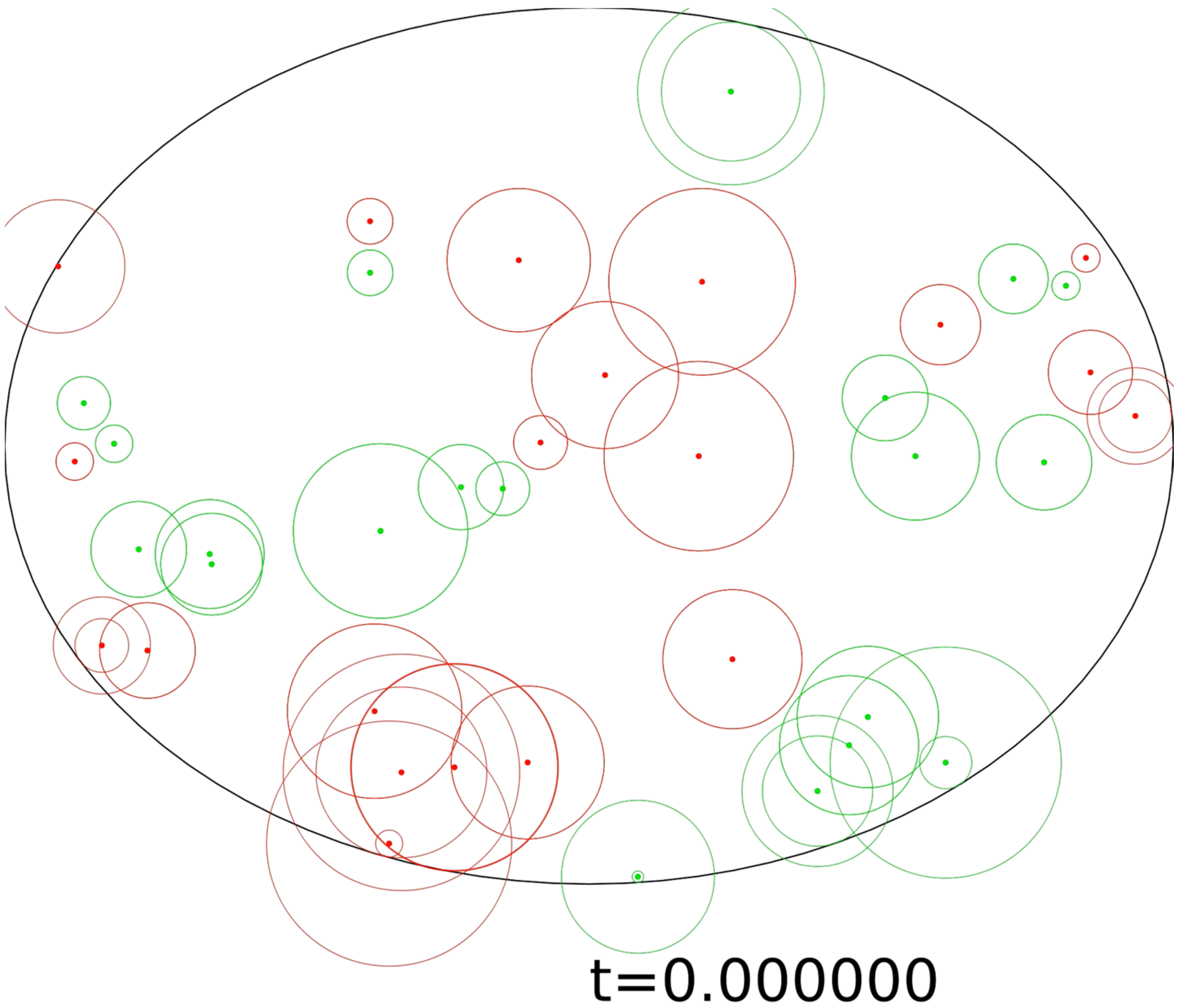


Demonstration of FPTMC in 2d:
Many reacting particles in a rectangle

Green particles on the left, red particles on the right, G+R→B



Demonstration of FPTMC in 2d:
Many reacting particles in an ellipse



RD dynamics with spatially varying reaction rate

Consider diffusion with spatially varying annihilation rate k(r,t):

$$\boxed{\frac{\partial P(\mathbf{r},t|\mathbf{r}_0,t_0)}{\partial t} = D\Delta P(\mathbf{r},t|\mathbf{r}_0,t_0) - k\left(\mathbf{r},t\right)P(\mathbf{r},t|\mathbf{r}_0,t_0)}$$

on domain G
with absorbing bc

Algorithm first passage times:

Input: $\mathbf{r}_0,\ t_0,\ t_{\max},\ k_m(t)$
Output: $\mathbf{r}, t$
$t \leftarrow t_0$
$\mathbf{r} \leftarrow \mathbf{r}_0$
repeat
  $t_a \leftarrow$ random number according to $\rho_m(\cdot|t)$
  $t_b \leftarrow$ random number according to $\rho_b^D(\cdot|\mathbf{r},t)$
  if $(t_{\max} < \min(t_a,t_b))$ **then**
    $\mathbf{r} \leftarrow$ random position according to $\rho_n^D(\cdot|t_{\max},\mathbf{r},t)$
    $t \leftarrow t_{\max}$
  else
    if $(t_a < t_b)$ **then**
      $\mathbf{r} \leftarrow$ random position according to $\rho_n^D(\cdot|t_a,\mathbf{r},t)$
    else
      $\mathbf{r} \leftarrow$ random position at the boundary ∂G
        according to $\rho_f^D(\cdot|t_b,\mathbf{r},t)$
    end if
    $t \leftarrow \min(t_a,t_b)$
  end if
until $\left(\left(\frac{k(\mathbf{r},t)}{k_m(t)} \geq \mathrm{ran}[0,1]\right) \text{ or } (t_a > t_b) \text{ or } (t = t_{\max})\right)$
return $(\mathbf{r},t)$

$k_m(t)$ = maximum of k(r,t)
t_{max} = maximum time

ρ_m = PDF of first annihilation times in domain G for homogeneous annihilation rate k_m

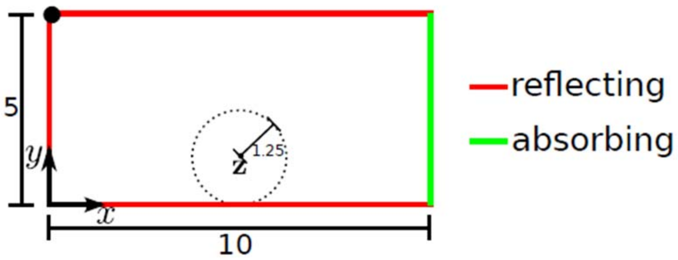
ρ_b = PDF of first passage times to domain boundary ∂G

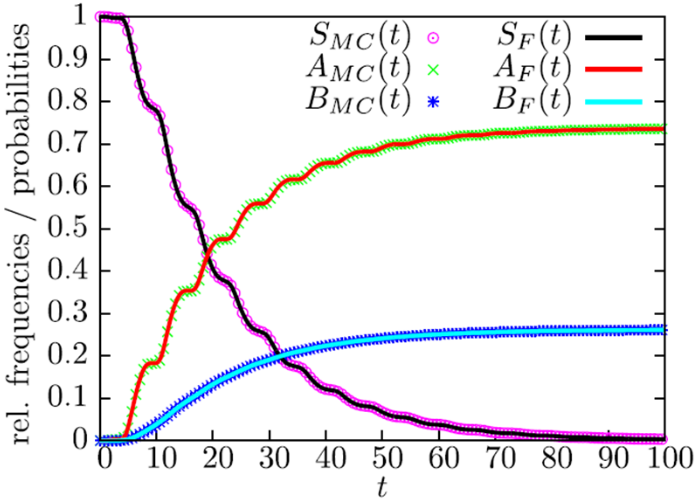
ρ_n = PDF of positions after free diffusion from t to t_a within G

ρ_f = PDF of positions on boundary ∂G after free diffusion from t to t_b

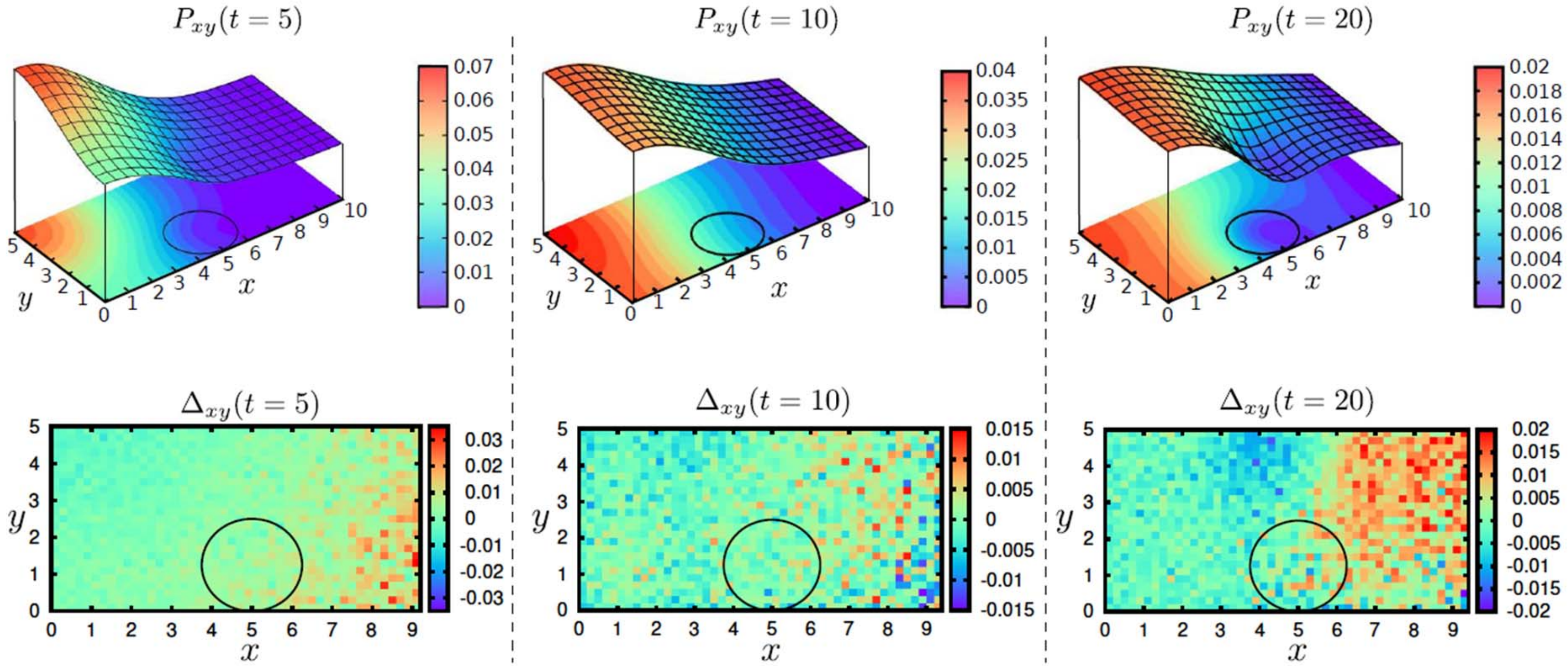
[Schwarz, Rieger – J. Comp. Phys. in press (2012)]

Example: Rectangle with oscillating annihilation zone





$$k(\mathbf{r},t) = \begin{cases} 3\left|\cos^3\left(\frac{t}{2}\right)\right| \cdot \left(c^2 - ||\mathbf{r}-\mathbf{z}||^2\right), & ||\mathbf{r}-\mathbf{z}|| < c \\ 0, & ||\mathbf{r}-\mathbf{z}|| \geq c \end{cases},$$



Literature

- M. A. Gibson and J. Bruck:
  Efficient exact stochastic simulation of chemical systems with many species and many channels, J. Phys. Chem. A 104, 1876 (2000)

- J. Elf and M. Ehrenberg:
  MesoRD - Mesoscopic Reaction Diffusion Simulator,
  http://mesord.sourceforge.net

- J. S. van Zon, P. R. ten Wolde:
  Green‘s function reaction dynamics: A particle based approach for simulating biochemical networks in time and space, J. Chem. Phys. 123, 234910 (2005)

- T. Oppelstrup, V. V. Bulatov, A. Donev, M. H. Kalos, G. Gilmer, B. Sadigh:
  First passage time kinetic Monte Carlo,
  Phys, Rev. E 80, 066701 (2009)

- K. Schwarz and H. Rieger:
  Efficient kinetic Monte Carlo method for reaction-diffusion problems with spatially varying annihilation rate,
  J. Comp. Phys. (in press), http://lanl.arxiv.org/abs/1206.2203