Kinetic Monte Carlo lecture

Initiation to the kMC methodology
Day 1

N. Salles

Kinetic Monte Carlo lecture

Goal of lecture:

- Understand the time evolution with probabilistic point of view
- Understand the structure the kMC algorithms
- Implementation of physical system
- Some optimization of algorithm

Material:

- kMC kernel code

Step_1 : Time introduction in Monte Carlo

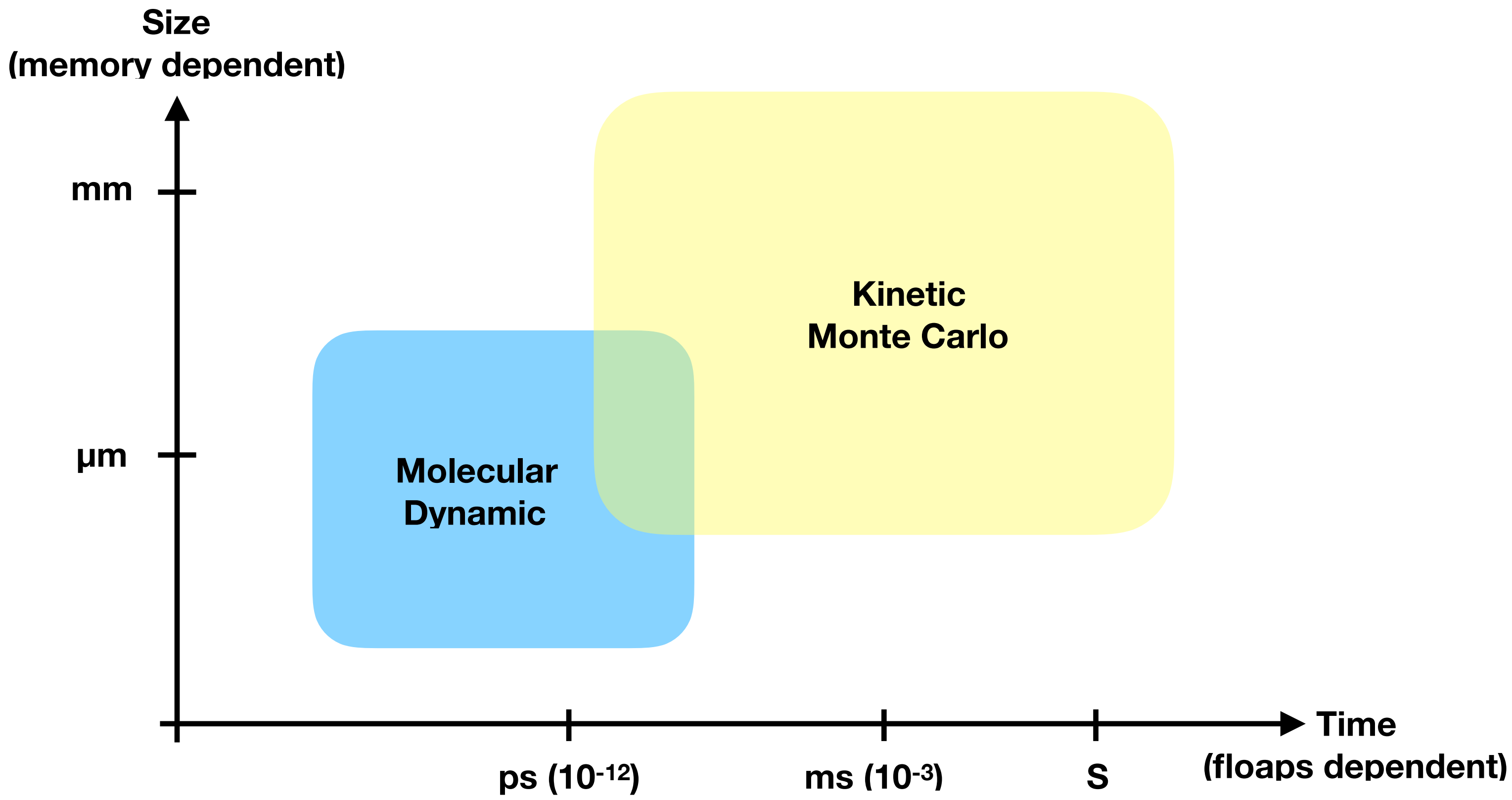
Step_2 : Some formula of rate

Introduction

- kinetic study of many problem in the Monte Carlo methodology frame
  - System of Spin
  - Solid growth
  - Chemical reaction
  - Electric current in solid
  - Biologic system

- Different algorithm exist
  - Time residence algorithm
  - First reaction algorithm

- Avantage is to be simple and permit to reach large time scale for large scale system



Time evolution simulation of physical system

From the master Equation

$$\frac{\partial P(\sigma,t)}{\partial t} = \sum_{\sigma'} r_{\sigma'\to\sigma} . P(\sigma',t) - r_{\sigma\to\sigma'} . P(\sigma,t)$$

At the steady state

$$r_{\sigma'\to\sigma} . P(\sigma',eq) = r_{\sigma\to\sigma'} . P(\sigma,eq)$$

In closed system $P(\sigma,eq) = \frac{1}{Z} . e^{-\frac{E(\sigma)}{k_B T}}$

$$r_{\sigma'\to\sigma} = r_{\sigma\to\sigma'} . e^{-\frac{E(\sigma)-E(\sigma^{'})}{k_B T}}$$

Metropolis Algorithm

σ more stable of $\sigma' \Rightarrow T(\sigma' \to \sigma) = 1$

σ less stable of $\sigma' \Rightarrow T(\sigma' \to \sigma) = e^{-\beta(E(\sigma)-E(\sigma'))}$

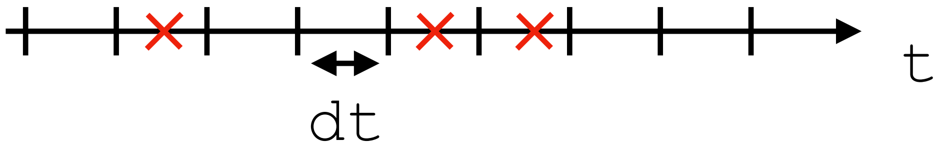
Monte Carlo = no time evolution

Time evolution simulation of physical system

Frequentist representation

t = n.dt,

- the time is divided in interval dt and
- some event happens during this time *t*



1) If we take *dt* enough small we can have, at most, one event by interval

2) we define the average rate:

$$r = \lim_{dt \to 0, t \to \infty} \sum^{n} \frac{n_{evt}}{t}$$ and $$p = r.dt$$

Time evolution simulation of physical system

Binomial representation

```
Binomial law b(n,p):
 - n the number of experiment
 - p (=r.dt) the probability to have a success
```

The probability to have x event in n trial:

$$P(X = x) = \binom{n}{x}(r.dt)^x(1 - r.dt)^{n-x}$$

Convergence to the Poisson law

We have n.p = n.r.dt = r.t > 0

```
In the limit of small dt:
 - n -> inf
 - p -> 0
 => The binomial law b(n,r.dt) converge to Poisson law of parameter r.t
```

$$P(X = x) = \frac{(rt)^x}{x!}e^{-rt}$$

The probability to have x event in n attempt
become -> the probability to have x event in time t

Time evolution simulation of physical system

Theorem:

- If $\mathbf{N_t}$ is number of event in time t and
- $\mathbf{T_n}$ is time between (n-1) and n-th event

Then $\mathbf{N_t}$ is Poisson process of parameter **K** if and only if $\mathbf{T_n}$ **are independent** and follow the exponential law with same parameter **K** and its density is:

Tn independent means that the event happens at time t is independent of event happened in previous time

Poisson law of parameter λ

$$P(X = x) = \frac{\lambda^x}{x!}e^{-\lambda}$$

$$< X > = \lambda$$

$$Var(X) = \lambda$$

Exponential law of parameter λ

$$P(X > t) = e^{-\lambda t}$$

$$< X > = \frac{1}{\lambda}$$

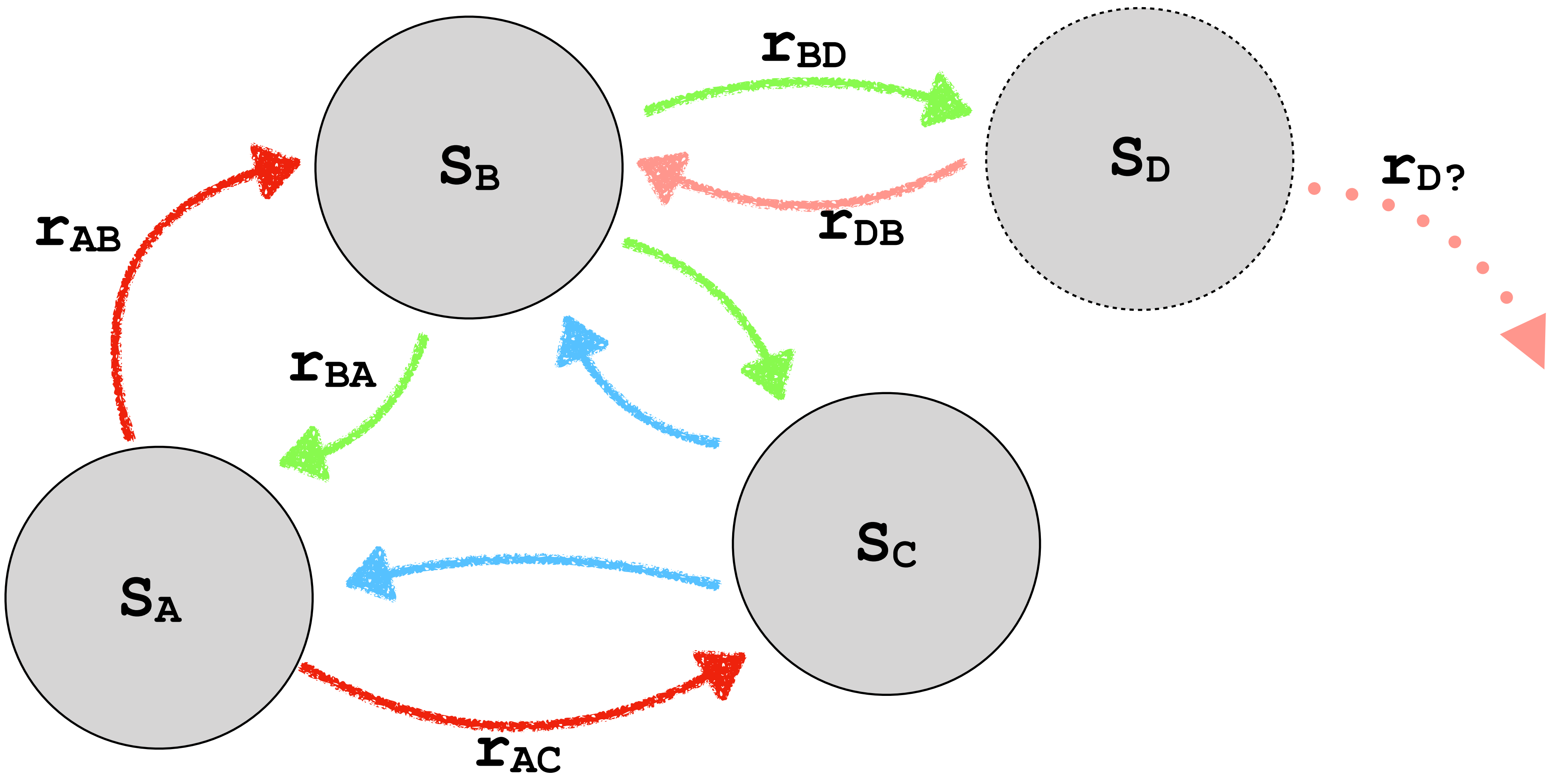
$$Var(X) = \frac{1}{\lambda^2}$$

Time evolution simulation of physical system

$$\frac{\partial P(\sigma,t)}{\partial t} = \sum_{\sigma'} r_{\sigma'\to\sigma} \,.\, P(\sigma',t) - r_{\sigma\to\sigma'} \,.\, P(\sigma,t)$$

=> The evolution time of system will be a Markovian chain of states S_i connected by the jump event between them

$$r_A = \sum_I r_{AI}$$



Chain: $S_A \xrightarrow[T_1]{} S_B \xrightarrow[T_2]{} S_A \xrightarrow[T_3]{} S_C \xrightarrow[T_4]{} S_B \xrightarrow[T_5]{} S_D \rightarrow \ldots$

$t=\Sigma T_i$

Time evolution simulation of physical system

T_n, time between events (n-1, n) => exponential law of parameter « r »

t_n: cumulative time at the event *n*

Probability that happens nothing in time $t_n+\tau$: $P(T_n > \tau) = e^{-r\tau} = P_0(\tau)$

Probability to have an event μ in next time dτ: $r_\mu d\tau$

We write the density of probability that happens an event between time τ and τ+dτ

$$P(\tau,\mu)d\tau = P_0(\tau)r_\mu d\tau \quad \longrightarrow \quad \int_0^\infty \sum_\mu^M P(\tau,\mu)d\tau = 1$$

We can also write: $P(\tau,\mu)d\tau = P_1(\tau)\,.\,P_2(\mu\,|\,\tau)\,.\,d\tau$

Happens an event in between time τ and τ+dτ

$$P_1(\tau)\,.\,d\tau = \sum_\mu^M P(\tau,\mu)\,.\,d\tau$$

Happens event μ given the next event occur at time τ

$$P_2(\mu\,|\,\tau) = \frac{P(\tau,\mu)}{\sum_\mu^M P(\tau,\mu)}$$

Time evolution simulation of physical system

1)Selection of event:

Happens event µ given the next event
occur at time τ

$$P_2(\mu \mid \tau) = \frac{r_\mu}{r}$$

Drawn an random number u_1~U[(0;1)] to choose an event µ:

$$\sum_i^{\mu-1} \frac{r_i}{r} < u_1 \leq \sum_i^{\mu} \frac{r_i}{r}$$

2) Determine Tn following an exponential law of parameter « r »:

$$F(x) = 1 - e^{-rx}$$

Repartition function of exponential law

We use the Inversion theorem to obtain the time as function of random number u2~U[(0;1)]:

$$T_n = \frac{\ln(1 - u_2)}{r} \sim \frac{\ln(u_2)}{r}$$

Time evolution simulation of physical system

In system or state S_i we identify all possible event E_{ij} with probability (or rate) r_{ij} to jump in state S_j

The total rate (or probability) to jump in other state S_{i+1} will be $r_i = \sum_j r_{ij}$

The random variable u follows the uniform distribution U[(0;1)]

Residence time algorithm (BKL)

1) List and calcul of rate of possible event in state S_i

2) Calcul of total rate r_i of the state S_i

3) Take a random variable u_1 to chose an event such as $\sum_j^{\mu-1} r_{i,j} < u_1 \, . \, r_i \leq \sum_j^{\mu} r_{i,j}$

4) Carry out event μ

5) Take a random variable u_2 to calcul the time elapsed dt since last event such as

$$dt = -\frac{\ln(u_2)}{r_i}$$

6) Increment the time t = t + dt

7) Back in step 1) if the criterium of loop is okay

Time evolution simulation of physical system

KMC Algorithm

An alternative algorithm can be build starting from each event point of view
We can write for each event possible in state S_i the density of probability for an event μ to occurs in intervalle time $(\tau, \tau+d\tau)$:

$$P_\mu(\tau)\,.\,d\tau = r_\mu\,.\,e^{-r_\mu.\tau}\,.\,d\tau$$

This is the density of probability of exponential law of parameter « r_μ ».

The partition function of process μ is $F(x) = 1 - e^{-r_\mu x}$

For each process we drawn u_μ~U[(0;1)]: $\tau_\mu = \dfrac{\ln(u_\mu)}{r_\mu}$

From this point of view we can find the previous density of probability as

$$P(\tau,\mu)d\tau = P_\mu(\tau)\,.\,d\tau\,.\,Prob\left\{\tau_\nu > \tau, \text{all } \nu \neq \mu\right\}$$

$$Prob\left\{\tau_\nu > \tau, \text{all } \nu \neq \mu\right\} = \prod_{\nu\neq\mu}^{M} e^{-r_\nu \tau}$$

$$P(\tau,\mu)\,.\,d\tau = r_\mu\,.\,e^{-\sum_\mu^M r_\mu.\tau}\,.\,d\tau$$

Time evolution simulation of physical system

KMC Algorithm

First-reaction algorithm (Gillepsie)

1) For each possible event E_{ij} in state S_i calcul the rate r_{ij} and take a random variable u_j. To calcul the « tentative reaction time »

$$dt_j = -\frac{\ln(u_j)}{r_{ij}}$$

2) Take the event which has the smaller « tentative reaction time » dt_j

3) Carry out event j

4) Increment the time t = t + dt

5) Back in step 1) if the criterium of loop is okay

Time evolution simulation of physical system

Resume

- The kinetic Monte Carlo simulation is a Markovian chain which the event are distributed with poisson law and the time between them follow the exponential law

- The event must selected as function of them probability

kinetic Monte Carlo:

- Versatile method, large kind of system can be describes
- Large system dimension & long time evolution

Difficulties:

- Description depend on the list of event
- Not fully Parallelizable but some optimization exist

Rate Calculation

Different way to compute the rate which are system dependent.

1) Historically, system of spin coupled with bath at temperature T:

$$\nu_0 = \frac{k_B \, . \, T}{h} \sim 2.10^{11} \, . \, T \quad s^{-1}$$

2) Adding particle from outside => FLUX

Particle from gaz:

Avogadro Number

Surface

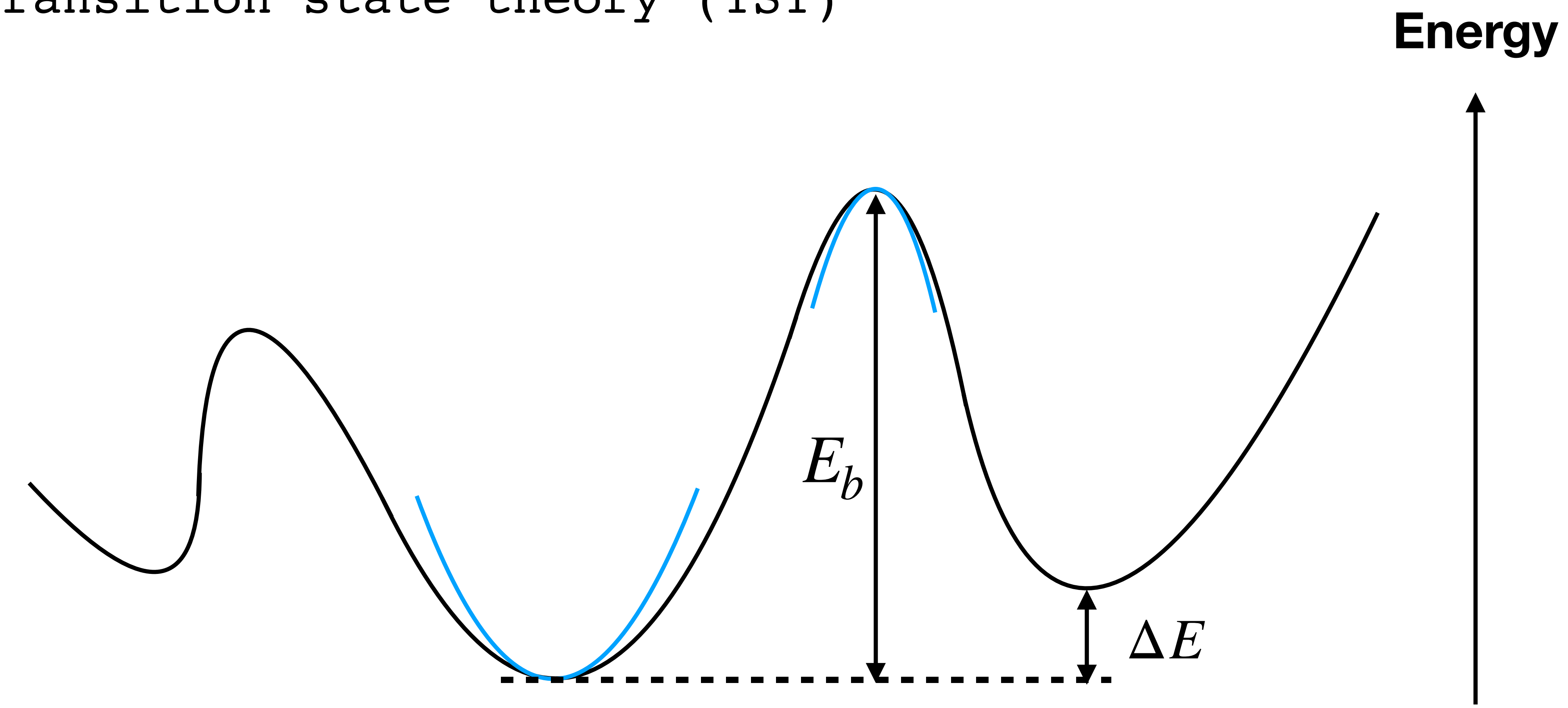
$$\mathscr{F} \, . \, S = \frac{P \, . \, \mathcal{N}_a \, . \, S}{\sqrt{2.\pi \, . \, M_p \, . \, R \, . \, T}}$$

Perfect gaz constante

Masse of particle

Rate Calculation

3) Chemical reaction and solid state physics
=> Transition state theory (TST)



$$r_i = \nu_0 \, . \, e^{-\frac{E_b}{k_B T}}$$

In solid $\nu_0 \sim 10^{12} - 10^{13} \, . \, s^{-1}$

Rate Calculation

4) In case of Chemical reaction we define:

$c_i dt$: The probability that a particular combination of R_i react in the next time interval dt

In certain condition, all the species are homogenous repartition of species in volume at each time, we can associate the reaction parameter $\mathbf{c_i}$ and the reaction rate constant $\mathbf{k_i}$:

$$k_i = Vc_i$$

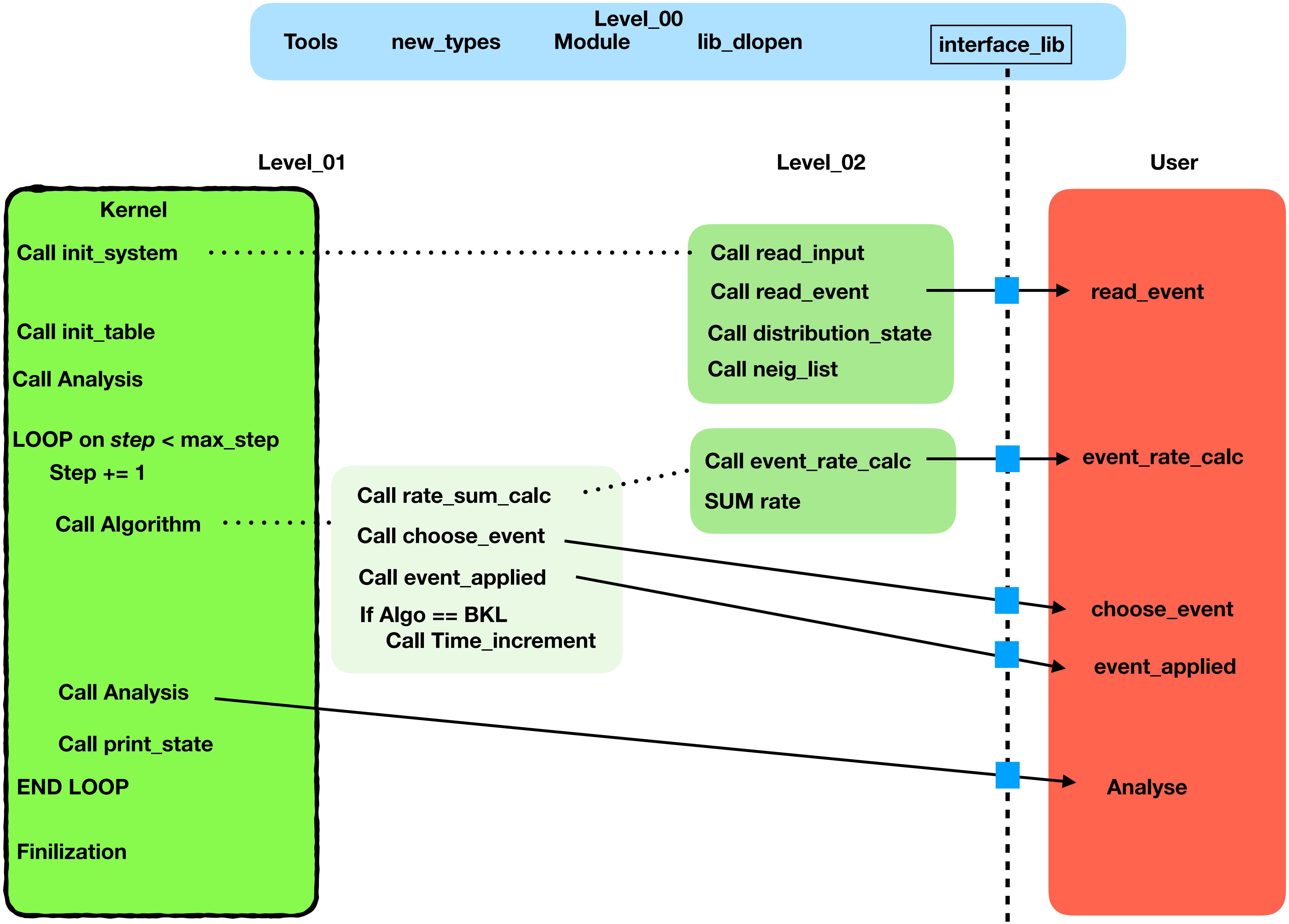
The rate $\mathbf{r_i=h_i.c_i}$ depend of the number of combination of species in reaction $\mathbf{R_i}$:

$$R_1 : X_1 + X_2 \to 2X_1 \;\Rightarrow\; h_1 = X_1 X_2$$

$$R_2 : 2X_1 \to X_1 + X_2 \;\Rightarrow\; h_2 = \frac{1}{2} X_1(X_1 - 1)$$

$$R_3 : 3X_1 \to 2X_1 + X_2 \;\Rightarrow\; h_3 = \frac{1}{6} X_1(X_1 - 1)(X_1 - 2)$$

kMC kernel code description



Level_00 Module

Tools new_types lib_dlopen interface_lib

Level_01

Kernel

```
Call init_system

Call init_table
Call Analysis

LOOP on *step* < max_step
     Step += 1

     Call Algorithm

     Call Analysis

     Call print_state
END LOOP

Finilization
```

```
Call rate_sum_calc
Call choose_event
Call event_applied
If Algo == BKL
    Call Time_increment
```

Level_02

```
Call read_input
Call read_event
Call distribution_state
Call neig_list
```

```
Call event_rate_calc
SUM rate
```

User

read_event

event_rate_calc

choose_event

event_applied

Analyse

kMC kernel code description

Two structure regroup all the importante arrays and variables

```
struct event_type {

  int nevent, nbond, nchem_react;

  int *ptr_i_state,   // array contains Initial state
      *ptr_f_state;   // array contains Final state

  double *ptr_f0,          // array contains prefactor rate
         *ptr_ebarrier,    // array contains activation energies
         *ptr_de,          // array contains difference energy between two miniums
         **ptr_ebond;      // 2D array contains bond energies
};
```

```
struct kmc_type {

  int bavard, conv;          // Verbose & convergence
  int period[3], nsites[3];  // Bound periodic condition & number of site

  char algorithm[50],        // algorithm name: BKL or Gillepsie
       input_file[50],       // input_file name
       input_event[50],      // event file name
       libname[50],          // shared lib name with the path
       init_mod[50];         // Initialization mode: species, random, none

  int tot_sites,             // Total number of sites
      max_step,              // max step criterium
      sys_dim,               // system dimension, 1, 2 or 3 D
      freq_write,            // write stat frequency
      nprop,                 // number of properties compute in analyse routine
      node_state,            // number of state of one site
      nspec,                 // number of different species
      npressure,             // number of partial pressure
      step;                  // actual step

  double sum_rate,           // Total rate of system
         rand_rate,          // random rate
         time,               // actual time
         rand_time,          // random time
         max_time,           // max time criterium
         temp,               // temperature value (Kelvin)
         kt,                 // Energy of temperature
         per100,             // Percentage of 0 state in initial system
         f0,                 // prefactor rate value
         scale;              // scale value

  int     *ptr_site, *ptr_nneig, *ptr_nevt, *ptr_neig, *ptr_event_site, *ptr_spec;
  double  *ptr_rate, *ptr_prop, *ptr_event_rate, *ptr_pressure, *ptr_masse;

  struct event_type event;
};
```