

Ασκήσεις Υπολογιστικής Φυσικής

Εξίσωση Master

Εισαγωγή

Γενική συνάρτηση επιλογής μετάβασης: Επιλέγει τυχαία ένα από τα `possible_results` με βάση τους ρυθμούς `transition_rates`. Επιστρέφει και το χρονικό διάστημα το οποίο χρειάστηκε για την μετάβαση.

```
# file: utils.py
import random
import numpy as np

def pick_transition(
    possible_results: np.array, transition_rates: np.array
):
    transition_intervals = transition_rates.cumsum()
    rate_sum = transition_intervals[-1] # last entry

    R = random.random() * rate_sum
    result_idx = (transition_intervals > R).nonzero()[0][0]

    dt = -np.log(random.random()) / rate_sum

    return (dt, possible_results[result_idx])
```

code block 0

Αμφίδρομη αντίδραση

Έστω A, X δυνατές καταστάσεις που μεταβαίνουν μεταξύ τους.

$$A \rightleftharpoons X$$

Ρυθμοί μετάβασης αν A ο (σταθερός)
αριθμός των A και N ο αριθμός των X :

$$W_{N+1,N} = k_1 A$$

$$W_{N-1,N} = k_2 N$$

Ρυθμοί μετάβασης αν A ο (σταθερός)
συνολικός αριθμός των A και X και N ο
αριθμός των X :

$$W_{N+1,N} = k_1 (A - N)$$

$$W_{N-1,N} = k_2 N$$

```
# file: two_way.py
import numpy as np
import utils

A = 20; N = 20
k1 = 1; k2 = 1

Tsim = 10000

t = 0
```

code block 1

Αμφίδρομη αντίδραση

```
# the number of A is constant
bins = 50
P = np.zeros(bins)

transitions = np.array([1, -1])
rates = np.array([k1 * A, k2 * N])
while t < Tsim:
    dt, dN = utils.pick_transition(
        transitions,
        rates
    )

    t += dt; N += dN
    rates[1] += k2 * dN

    try: P[N] += dt
    except IndexError: pass

P /= t
```

code block 2

```
# the number of A is not constant
bins = 20
P = np.zeros(bins)

transitions = np.array([1, -1])
rates = np.array([k1*(A-N), k2 * N])
while t < Tsim:
    dt, dN = utils.pick_transition(
        transitions,
        rates
    )

    t += dt; N += dN
    rates[0] -= k1 * dN
    rates[1] += k2 * dN

    try: P[N] += dt
    except IndexError: pass

P /= t
```

code block 3

Αμφίδρομη αντίδραση

