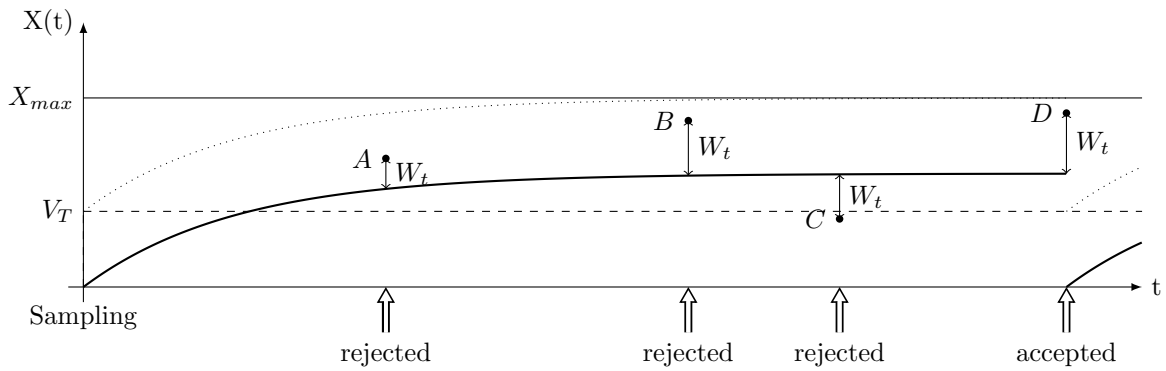
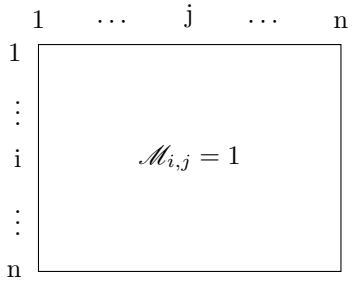
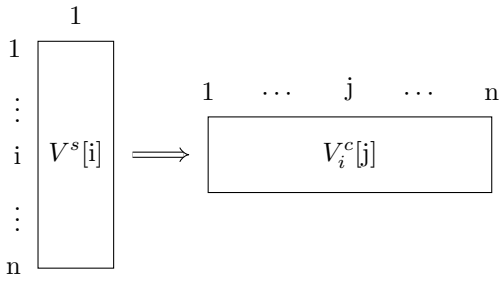


Figure 1: Rejection sampling procedure for a stochastic system with random noise W_t



(a) Matrix of children $\mathscr{M}_{i,j} = 1$ if j is children of i and 0 otherwise.



(b) Vector of seeds and reconstructed vector of children

```
1: RNG: a random number generator
2: p: probability of connection between two neurons
3: M_{i,j}: matrix of interaction
4: function InteractionMatrix(RNG, p)
5:     for all (i, j) ∈ {1, ··· , n}^2 do
6:         M_{i,j} ← RNG.B(p)
```

(c) Generation of a matrix of children

```
1: RNG: a random number generator
2: p: probability of connection between two neurons
3: V^s: vector of states
4: V_i^c: vector of children of neuron i
5: function Reconstructible(RNG, p)
6:     for i ← 1 to n do
7:         V_i^s ←RNG.State
8:         for j ← 1 to n do
9:             RNG.B(p)                ▷ Generate, without storing them, the children of i
10:    return V_i^s
11: function Reconstruct(RNG, i, V^s)
12:    RNG.setState(V^s[i])
13:    for j ← 1 to n do
14:        V_i^c[j] ←RNG.B(p)
15:    return V_i^c
```

(d) Generation of a vector of rng states

Figure 2: Algorithm for generating erdos-reynii graph in memory (a and c) and reconstructible (b and d)

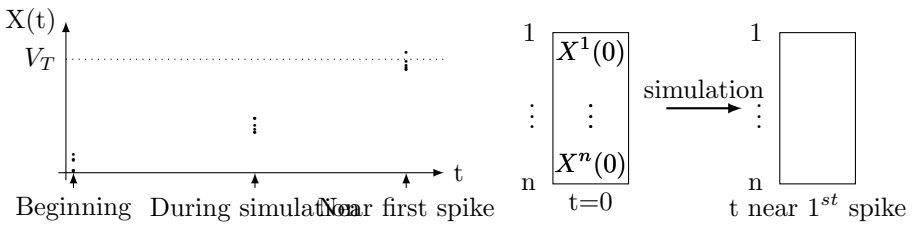


Figure 3: Illustration of the initialisation process, the state of the system is represented at different instants of the simulation (each dot is a neuron, x-abscissa time and y-abscissa the potential of the neuron).

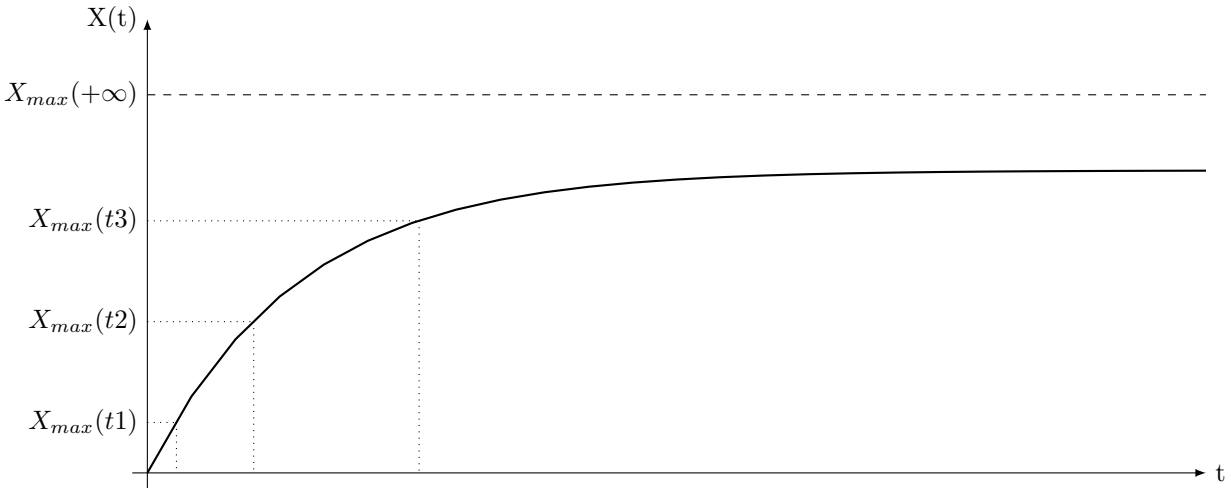


Figure 4: Restriction of time interval searching for the rejection sampling procedure