Markov Chains, Iterated System of Functions and Coupling time for Perfect Simulation

Jean-Marc Vincent*

Abstract

Simulation of Markov chains are usually based on an algorithmic representation of the chain. This corresponds to stochastic recurrent equation and could be interpreted as random iterated systems of functions (RISF). In particular, for perfect simulation of Markov chains, the RISF structure has a deep impact on execution time of the simulation. Links between the structure of the RISF and coupling time of algorithm are detailed in this paper. Conditions for coupling and upper bound for simulation time are given for Doeblin matrices. Finally, it is shown that aliasing techniques build an RISF with a particular binary structure.

1 Introduction

Markov chains are basic tools to study random dynamical systems. They play the central role of linear part of the dynamic and capture most of the dynamic characteristics. When the system is finite, the Markov chain is described by its transition kernel (stochastic matrix). When the system is homogeneous in time, irreducible and aperiodic, the left eigenvector π associated to the eigenvalue 1 captures most of informations needed in practical applications. Difficulties arise when the size of the system is too large so that traditional linear algebra tools could not be used.

For a large state space, simulation provides methods based on an algorithmic representation of the chain and offers new possibilities for the statistical estimation of π. Unfortunately, these methods are empirical and the management of errors is very difficult. Perfect simulation techniques Propp and Wilson (1996) have been developed in the last 10 years. These methods guarantee the convergence to steady-state in a finite number of steps and help for the simulation control.

In fact, the algorithmic representation of the Markov chain could be interpreted as a random iterated systems of functions. The aim of this article is to investigate relations between the Markov chain and its representations as RISF. It is shown that the RISF impacts deeply the simulation time for perfect simulation.

In the second section the RISF formalism is introduced and properties of Markov chains from a RISF are deduced. Then forward simulation and its drawbacks is presented. Section

*This work was partially supported by ACI SurePath and ANR SMS

5 introduces the perfect simulation algorithm and the convergence criteria in a finite number of steps. Relations between the RISF and coupling time are given in section 6 and applied for Doeblin matrice (section 7). Finally the alias technique is described and the uniform-binary decomposition is proposed.

2 Random Iterated System of Functions

Consider a finite state space identified to the set $\mathcal{X} = \{1, \cdots, K\}$ and consider a finite family of m functions $\mathcal{F} = \{f_\theta : \theta \in \Theta\}$ that maps $\mathcal{X}$ onto itself. Denote by F_θ the matrix of the operator f_θ with K rows and K columns,

$$F_\theta(i,j) = \begin{cases} 1 & \text{if } f_\theta(i) = j; \\ 0 & \text{if not.} \end{cases}$$

A probability distribution $\mathcal{P} = \{p_\theta : \theta \in \Theta\}$ is given on the set Θ of functions and the dynamic of the system is defined by

$$X_0 = x_o, \; X_1 = f_{\theta_1}(X_1),$$

$$X_{n+1} = f_{\theta_{n+1}}(X_n) = f_{\theta_{n+1}} \circ f_{\theta_n} \circ \cdots f_{\theta_1}(X_0), \tag{1}$$

where $\{\theta_n\}_{n\in\mathbb{N}}$ is a random sequence of elements of Θ chosen independently according to distribution $\mathcal{P}$. It is clear that the stochastic process $\{X_n\}_{n\in\mathbb{N}}$ is a homogeneous discrete time Markov chain, because the conditional distribution of the future does not depend on the past.

In the domain of probability such stochastic recurrence equations have been widely studied when the state space is continuous Diaconis and Freedman (1999) or from an ergodicity point of view Borovkov and Foss (1994, 1992); Stenflo (1998). More general results are developed in Meyn and Tweedie (1993) and Brémaud (1999).

Because the state space is finite, the dynamic of the process is given by the transition matrix P of the Markov chain. It is obtained by

$$p_{i,j} \overset{\Delta}{=} \mathbb{P}(X_{n+1} = j | X_n = i) = \sum_{\theta\in\Theta} p_\theta . 1\!\!1_{f_\theta(i)=j}.$$

This transition matrix is non-negative and the sum of elements on a row equals 1.

The irreducibility of the Markov chain is related to pattern properties of functions.

Proposition 2.1 (Irreducibility)**.** *Suppose that for each couple (i,j) of states there exists a finite pattern $(\theta_1, \cdots, \theta_l)$ such that the probability of the pattern is positive $p_{\theta_1} \cdots p_{\theta_l} > 0$ and*

$$j = f_{\theta_l} \circ f_{\theta_{l-1}} \circ \cdots f_{\theta_1}(i),$$

then the Markov chain associated to the RISF is irreducible.

Moreover, aperiodicity of the chain is deduced from the support of functions f_θ.

Proposition 2.2 (Aperiodicity)**.** *The irreducible Markov chain is aperiodic if for each couple (i, j) there exist some n_0 such that for every $n \geq n_0$ there is a sequence $(\theta_1, \cdots, \theta_n)$ such that*

$$j = f_{\theta_n} \circ f_{\theta_{n-1}} \circ \cdots f_{\theta_1}(i).$$

One should note that if the Markov chain is irreducible and aperiodic there exists n_0 such that P^{n_0} is positive. Then the central convergence theorem from Kolmogorov, extension of the Perron-Froebenius in the finite case, could now be reformulated using random iterated system of functions:

Theorem 2.3 (Kolmogorov)**.** *If the RISF is aperiodic and irreducible (recurrent positive), then there exist a unique probability measure $\pi = (\pi_1, \cdots, \pi_K)$ (line vector) satisfying*

$$\pi = \pi P = \sum_{\theta} p_\theta \pi F_\theta, \tag{2}$$

and for all (i, j)

$$\lim_{n \to +\infty} \mathbb{P}(f_{\theta_n} \circ f_{\theta_{n-1}} \circ \cdots f_{\theta_1}(i) = j) = \pi_j.$$

Intuitively, when we stop a RISF after a long period, the probability that the observed value is j is approximatively π_j. Moreover, if we compute the proportion of steps spent state j, this proportion converges to π_j (ergodic theorem).

Theorem 2.4 (Ergodic theorem)**.** *When the RISF is recurrent positive, then the Cesaro limit converges :*

$$\lim_{n \to +\infty} \frac{1}{n} \sum_{k=1}^{n} \mathbb{1}_{f_{\theta_n} \circ f_{\theta_{n-1}} \circ \cdots f_{\theta_1}(i) = j} = \pi_j \text{ almost surely.}$$

Theorem 2.3 guarantees that sampling independent sufficiently long trajectories gives an estimate of the stationary distribution. The ergodic theorem 2.4allows sampling on only on a single trajectory because the probability that the Cesaro limit does not converge is 0.

3 Estimation of π

When the size of the system is sufficiently small, formal or numerical computations provide the eigenvector π of the transition matrix P. If the size of the state space is too large, a simulation builds an estimate of π. The first technique called forward simulation is based on theorem 2.3 and leads to algorithm 3.1.

Algorithm 3.1. Forward simulation (independent sample generation)

```
  $n = 0$;
  $x = x_0$;
  {choice of the initial state at time n=0}
  **repeat**
```

```
    $n = n + 1$;
    $f_\theta = Random\_function()$;
    {Random function chosen according to the distribution $\{p_\theta : \theta \in \Theta\}$}
    $x = f_\theta(x)$;
    {computation of the next state $X_{n+1}$}
  **until** n = simulation length
  return $x$
```

This algorithm returns a state and we hope that, for a sufficiently long simulation run, the returned state distribution is a good approximation of π. So repeating the algorithm, we get a sample of independent realizations of π distributed random variables.

The problem of this approach is first the estimation of the simulation length (stabilization time or burn-in time). In usual softwares, this value is fixed empirically by the user. Moreover, because we generate a sample of independent variables the convergence of estimates of π converges very slowly to the limit value, in the order of $\mathcal{O}(\frac{1}{\sqrt{n}})$.

This algorithm could be extended by making sampling directly on the trajectory (ergodic sampling). In that case, samples are not independent and we should assume mixing properties of the process to justify the speed of convergence. Moreover, it has been shown that the convergence to steady-state depends on the spectral gap of the matrix P (module of the difference between 1 the first eigenvalue and the second eigenvalue).

Suppose now given a transition matrix P, the problem is to compute or estimate the steady state distribution π. The classical method consists in three steps :
(1) Build a set of functions $\mathcal{F} = \{f_\theta : \theta \in \Theta\}$ and the corresponding probabilities $\mathcal{P} = \{p_\theta : \theta \in \Theta\}$ such that $P = \sum_\theta p_\theta F_\theta$;
(2) Simulate a sample by algorithm 3.1;
(3) Estimate statistics on the sample.

Proposition 3.2. *The convergence of the forward algorithm does not depend on the set of function $\mathcal{F}$ and $\mathcal{P}$.*

This is clear because the construction of $\mathcal{F}$ does not modify the matrix P and so convergence to steady state. Usually, the construction of the family of functions $\mathcal{F}$ and $\mathcal{P}$ are based on randomized algorithms. Because the set of states is finite, usual algorithms are like the following 3.3 (inverse probability distribution function):

Algorithm 3.3. Next state generation

```
  {Current state is $i$}
  $s = 0$; $j = 1$;
  $u = Random(0, 1)$;
  **while** $u > s$ **do**
    $s = s + P[i, j]$; $j = j + 1$;
  **end while**
  return $j$
```

This representation leads to set $\mathcal{F}$ with cardinality at most $m - (K-1)$ where m is the number of positive elements of P. This algorithm could be improved, by tree structures or hash tables, the main idea is to consider an appropriate segmentation of the interval $[0, 1[$ as shown in section 8.

4 Forward coupling

An intuitive idea (not so good as shown in example 1), to stop simulation is to consider all possible initial values, observe their trajectories and stop the simulation when they are all in a same state. We say that all trajectories have coupled. The coupling time is the first time when the trajectories are all in the same state, after the coupling time, the trajectories do not depend on the initial state. The recurrent equation (1) is applied to each state of $\mathcal{X}$ and we denote by $y(x)$ the current value of the trajectory issued from state x.

Algorithm 4.1. Forward-coupling simulation

```
for all $x \in \mathcal{X}$ do
  $y(x) = x$;
  {choice of the initial value of the vector $y$, $n = 0$}
end for
repeat
  $n = n + 1$;
  $f_\theta = Random\_function()$;
  {Random function chosen according to the distribution $\{p_\theta : \theta \in \Theta\}$}
  for all $x \in \mathcal{X}$ do
    $y(x) = f_\theta(y(x))$;
    {computation of the next state of the trajectory issued from $x$ at time 0}
  end for
until All $y(x)$ are equal
return $y(x)$
```

An example of a forward-coupling simulation is illustrated by figure 1.

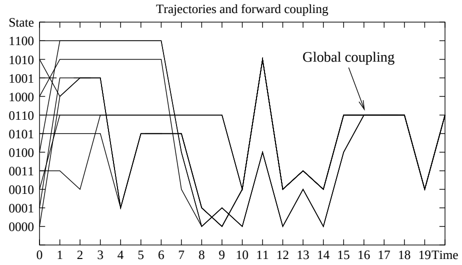
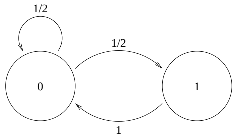


Figure 1: All trajectories have coupled before time $\tau^* = 16$

When the forward coupling algorithm stops all trajectories have coupled, unfortunately the generated state does not follows the stationary distribution. This is clearly illustrated in the example 1.

Example 1 : Coupling in a same state



On the example to the left, it is clear that coupling does not depend of the representation and that coupling time is almost surely finite, geometrically distributed with parameter $\frac{1}{2}$. When 2 trajectories couple, at the preceding step the corresponding states were 0 and 1. But, because the transition probability from 1 to 1 is zero, the trajectories can only couple in 0. Then the generated state is always 0, and is not distributed according to the stationary distribution $\pi = [\frac{2}{3}, \frac{1}{3}]$.

5 Backward Simulation Scheme

To make this algorithm "exact", Propp and Wilson (1996) propose to shift the process in the past. This is equivalent to Loynes (1962) monotone scheme used to prove the law convergence of the workload of a queuing system.

Provided that the representation of the Markov chain ensures coupling, we modify the algorithm (4.1) by reversing time leading to algorithm 5.1:

Algorithm 5.1. Backward-coupling simulation

```
for all $x \in \mathcal{X}$ do
  $y(x) \leftarrow x$ {choice of the initial value of the vector $y$, $n = 0$}
end for
repeat
  u $\leftarrow$ Random; {generation of $f_{\theta_{-n}}$}
  for all $x \in \mathcal{X}$ do
    $y(x) \leftarrow y(f_{\theta_{-n}}(x))$; {computation of the state at time 0 of the trajectory issued from $x$ at time $-n$}
  end for
until All $y(x)$ are equal
return $y(x)$
```

We illustrate the behavior in figure (2).

To understand this algorithm and find conditions for termination, we consider the sequence of subsets of the state space $\mathcal{X}$, $\{\mathcal{Z}_n\}_{n\in\mathbb{N}}$ defined by

$$\mathcal{Z}_n = f_{\theta_{-1}} \circ f_{\theta_{-2}} \circ \cdots f_{\theta_{-n}}(\mathcal{X}). \tag{3}$$

Because $f_{\theta_{-n}}(\mathcal{X}) \subset \mathcal{X}$, we deduce that the sequence $\{\mathcal{Z}_n\}_{n\in\mathbb{N}}$ is non-increasing. Using the finiteness of $\mathcal{X}$ and monotonicity, we obtain that $\{\mathcal{Z}_n\}_{n\in\mathbb{N}}$ converges almost surely to a set $\mathcal{Z}_\infty$. The system is coupling if $\mathcal{Z}_\infty$ is reduced to one point.

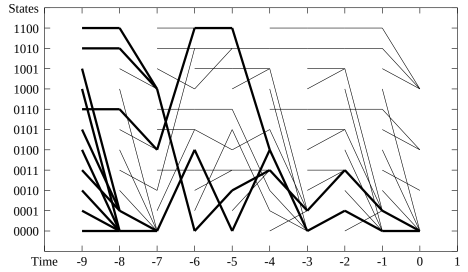


Figure 2: All trajectories collapsed in state 0000 after 9 steps

The next theorem, Propp and Wilson (1996); Vincent and Marchand (2004), states the fundamental result of the method:

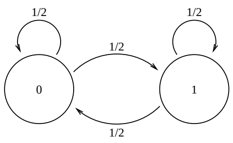
Theorem 5.2. *Suppose that the algorithm terminates, then the generated value $y(x)$ by algorithm 5.1 is distributed according to the stationary distribution.*

The difficulty is now to obtain conditions under which the coupling time is finite almost surely. An example which illustrate this difficulty is given in section 6 and construction of RISF ensuring that the algorithm terminates are detailed in section 7 and 8.

6 Coupling time

It was shown in proposition3.2 that the choice of the RISF implementing the Markov chain did not affect the convergence of the process. In the backward coupling scheme the situation is clearly different. Consider the following example of a two states Markov chain, in which we suppose that the$\{U_n\}$ are uniformly distributed on $[0, 1]$.
Example 2 : Three representations of a same Markov chain

This example, as example 1 before, is derived from the course of (Häggström, 2002, chap. 10). Consider the simple two states Markov chain :



Its transition matrix is

$$P = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix},$$

the chain has a unique stationary distribution

$$\pi = (\frac{1}{2}, \frac{1}{2}).$$

The three following RISF $f = (f_1, f_2)$ with probability $p_f = (\frac{1}{2}, \frac{1}{2})$, $g = (g_1, g_2)$ with probability $p_g = (\frac{1}{2}, \frac{1}{2})$, and $h = (h_1, h_2, h_3, h_4)$ with probability

$p_h = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ whose values are given by

$$\left\{ \begin{array}{l} f_1(0) = f_1(1) = 0, \\ f_2(0) = f_2(1) = 1; \end{array} \right. \left\{ \begin{array}{l} g_1(0) = g_2(1) = 0, \\ g_1(1) = g_2(0) = 1; \end{array} \right. \quad \text{and}$$

$$\left\{ \begin{array}{l} h_1(0) = h_1(1) = h_2(0) = h_3(1) = 0, \\ h_2(0) = h_3(0) = h_4(0) = h_4(1) = 1, \end{array} \right.$$

represent the same transition matrix P. The coupling time for the first representation equals 1, for the second representation the algorithm never terminates and the coupling time is geometrically distributed with mean 2 for the last case.

This example shows that the construction of the RISF is of crucial importance on the coupling time. We observe that the behavior of the system could exhibit mean coupling time arbitrarily large and in some cases the coupling time is infinite.

7 Doeblin matrices

Consequently, to avoid this problem, the idea is to build the RISF in order to ensure coupling. Constant functions are of great interest as :

Proposition 7.1. *If there exist a constant function$f_\theta \in \mathcal{F}$ then the coupling time is almost surely finite and stochastically bounded above by a geometric distribution with parameter p_θ.*

When the function f_θ is picked, all trajectories collapse in one state and coupling occurs. The coupling time is dominated by the first occurrence time of f_θ in the independent process of $\{f_{\theta_n}\}$ which is geometrically distributed.

When the transition matrix is positive (Doeblin matrix), it is possible to build a RISF which could couple in each state in just one step. Denote by

$$\alpha_j = \min_i p_{i,j}.$$

The Doeblin condition is simply

$$\min_j \alpha_j > 0. \tag{4}$$

Consequently the the algorithm 7.2 compute the next step of the chain and ensures coupling.

Algorithm 7.2. Next states generation (Doeblin matrices)

```
$u = Random(0,1)$;
if $u < \sum \alpha[j]$ then
  $s = 0$; $j = 0$;
  while $u > s$ do
    $j = j + 1$; $s = s + \alpha[j]$;
  end while
  for $i = 1$ to $n$ do
```

```
      $NextState(i) = j$
    end for
  else
    for $i = 1$ to $n$ do
      $s = \sum \alpha[j]$; $j = 0$;
      while $u > s$ do
        $j = j + 1$;$s = s + P[i, j] - \alpha[j]$;
      end while
      $NextState(i) = j$
    end for
  end if
```

Proposition 7.3. *The RISF represented by algorithm 7.2 couples in finite time and the mean coupling time is less than*

$$\frac{1}{\sum_{i=1}^{K} \alpha_i}$$

In fact this proposition is still valid and the algorithm still works if there is a column of positive elements, so inequality 4 is replaced by the weaker inequality:

$$\max_i \left\{ \min_j p_{i,j} \right\} > 0.$$

This condition implies that there exists a state that is accessible from every other state in $\mathcal{X}$. But, usually this condition is too strong for classical Markovian models for which the transition matrix is sparse.

8 Uniform-Binary decomposition

The aliasing technique, designed by Walker (1974), provides an efficient method to build the set of functions $\mathcal{F}$, that simulates the next state following i according to the transition probability $\{p_{i,j}\}_{j \in \mathcal{X}}$. Compared to classical methods Bratley et al. (1983) such as inverse of probability distribution function (algorithm 3.3), rejection, or composition methods, the complexity of the computation of the next state is in $\mathcal{O}(1)$, and so does not depend on the problem size.

Consider a typical distribution $q = (q_1, \cdots, q_K)$ on K states. The idea is to build a set of K thresholds $\{s_1, \cdots, s_K\}$ $0 \leq s_i \leq 1$, and K couples of states $\{(i_1, i'_1), \cdots, (i_K, i'_K)\}$ with $(i_j, i'_j) \in \mathcal{X}^2$, i'_j is called the alias value of i_j. This construction should verify the following constraints :

$$\forall i \in \{1, \cdots, K\}, \;\; q_i = \sum_{j=1}^{K} \left(s_j \mathbb{1}_{i_j = i} + (1 - s_j) \mathbb{1}_{i'_j = i} \right). \tag{5}$$

Such a decomposition is built by a simple algorithm requiring $\mathcal{O}(K)$ steps. The implementation structure is described in Bratley et al. (1983). From this structure the simulation runs as follows :

Algorithm 8.1. Aliasing generation

```
{ The values of s, and couples $(i_l, i'_l)$ are preliminary stored in arrays of size $K$ : $S$, $I$ and $I'$.}
$u = Random(0,1)$;
$v = Random(0,1)$;
$l = int(u * k)$; { discrimination among $K$, int means the integer part }
**if** $v < S[l]$ **then**
   return $I[i]$ { the standard value}
**else**
   return $I'[i]$ { the alias value}
**end if**
```

One should notice that this representation is not unique and, according remarks on the impact of representation on coupling time, we should use heuristics to build a "better" representation. In particular very interesting property of such a construction is that any permutation of two couples $(s_j, (i_j, i'_j))$ and $(s_l, (i_l, i'_l))$ provide another random variable with exactly the same distribution. Moreover, if we replace some threshold s_j by $1 - s_j$ and exchange the values (i_j, i'_j) to (i'_j, i_j), the distribution of the result is also preserved.

Consequently, we have two steps in the computation of the simulation kernel. A first step compute for each state x the corresponding arrays S_i, I_i, and I'_i (cf algorithm 8.1). The second step modify these arrays to guarantee termination in a finite number of steps. To simplify this step, we suppose that there exist some state i_0 such that the transition probability from i_0 to i_0 is strictly positive. This condition is stronger than aperiodicity and is generally verified in practical situations. If not, the matrix $\frac{1}{2}(Id + P)$ exhibits the same stationary distribution as P and could be used instead of P.

Because the Markov chain is irreducible there exist a spanning tree of the state space graph such that a path of positive probability exists in the tree from each state i to i_0. Because each state has an out-degree of 1 in the tree, it is always possible to place the next state of i in the tree (on the path to i_0) in the place $I_i[0]$. Let $\alpha = \min_i \frac{S_i[0]}{d_i}$, with d_i the out-degree of state x in the graph. α is strictly positive and with probability α all the transitions occur on the arrows of the tree. Repeating this transition K times leads to a global coupling in state i_0. The algorithm terminates almost surely. Moreover, if we denote by D the depth of the tree, the coupling time is upper bounded by a geometric distribution with parameter α^D, probability that a burst of D sequential transitions occur on the tree.

The aliasing technique remains valid if some probability are 0 in the distribution q, the corresponding threshold equals 0. Consider that all the alias computations are done on $\{1, \cdots, K\}$ and denote by S and A the threshold and the alias matrices.

Algorithm 8.2. Next states generation (Alias matrices)

```
$u = Random(0,1)$;
$k = int(u * K) + 1$;{ choice of the column}
$v = Random(0,1)$;
**for** $i = 1$ to $n$ **do**
```

```
    if  $v < S[i,k]$ then
      $NextState(i) = k$ { the standard value}
    else
      $NextState(i) = A[i,k]$ { the alias value}
    end if
  end for
```

For this situation, it appears that the transition matrix have been decomposed in a sum of K stochastic matrices

$$P = \frac{1}{K}\left(P_1 + \cdots + P_K\right),$$

where the stochastic matrix P_i have at most two non null elements per row. It corresponds to very simple structures which need further research. The number of matrices in the decomposition could easily be reduced to the maximum out degree of the chain d_{max}, which is useful for sparse matrices. In that case the cardinal of $\mathcal{F}$ is at most $(K+1).d_{max}$.

Moreover using permutations of columns or thresholds could improve the coupling time. The problem of finding the best Uniform-binary RISF minimizing the mean coupling time seems to be very complex. Some heuristics have been developed but are not yet published.

9 Conclusion

In practical examples, the RISF representation of a Markov chain is crucial for simulation. Several methods have been implemented in a software PSI [1] in order to test and compare heuristics. All results show that the problem is very difficult and needs further fundamental research.

Experiments on practical performance evaluation problems Vincent (2005) and Vincent and Marchand (2004) show that Markov chains with up to one million of states could be simulated by this technique. The amount of memory used by the algorithm (alias tables, threshold) is about 2 times the number of positive elements of the transition matrix. Then simulation time is sufficient to estimate parameters on the model typically scala products of a reward vector and the stationary distribution.

When the system is monotone, and that is the case in many practical situations of performance evaluation Vincent (2005) or in interacting systems of particles Propp and Wilson (1996), the method could be adapted by driving simulation from maximal and minimal states. The coupling time estimation still remains open.

References

A.A. Borovkov and S. Foss. Stochastically recursive sequences and their generalizations. *Siberian Advances in Mathematics*, 2(1), 1992.

[1] `http://www-id.imag.fr/Logiciels/psi/`

A.A. Borovkov and S. Foss. Two ergodicity criteria for stochastically recursive sequences. *Acta Appl. Math.*, 34, 1994.

P. Bratley, B.L. Fox, and L.E. Schrage. *A Guide to Simulation.* Springer-Verlag, 1983.

P. Brémaud. *Markov Chains: Gibbs fields, Monte Carlo Simulation and Queues.* Springer-Verlag, 1999.

P. Diaconis and D. Freedman. Iterated random functions. *SIAM Review*, 41(1):45–76, 1999.

O. Häggström. *Finite markov chains and algorithmic applications.* Cambridge University Press, 2002.

R.M. Loynes. The stability of queues with non independent inter-arrival and service times. *Proc. Cambridge Ph. Soc.*, 58:497–520, 1962.

S.P. Meyn and R.L. Tweedie. *Markov chains and stochastic stability.* Communications and Control Engineering Series. Springer-Verlag, 1993.

J. Propp and D. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9(1&2):223–252, 1996.

O Stenflo. *Ergodic theorems for Iterated Function Systems controlled by stochastic sequences.* Doctoral thesis n. 14, Umea university, 1998.

J.-M. Vincent. Perfect simulation of queueing networks with blocking and rejection. In *Saint*, pages 268–271, 2005.

J.-M. Vincent and C. Marchand. On the exact simulation of functionals of stationary markov chains. *Linear Algebra and its Applications*, 386:285–310, 2004.

A.J. Walker. An efficient method for generating discrete random variables with general distributions. *ACM Trans. Math. Software*, 3:253–256, 1974.

Jean-Marc Vincent
Laboratoire ID-IMAG,
MESCAL Inria project,
51, avenue Jean Kuntzmann, F-38330 Montbonnot, France
Jean-Marc.Vincent@imag.fr
http://www-id.imag.fr/Laboratoire/Membres/Vincent_Jean-Marc/