Simulating Events of Unknown Probabilities via Reverse Time Martingales

K. Latuszynski*   I. Kosmidis

O. Papaspiliopoulos   G.O. Roberts

• Let $s \in [0,1]$ and $G_0 \sim U(0,1)$

- Let $s \in [0,1]$ and $G_0 \sim U(0,1)$

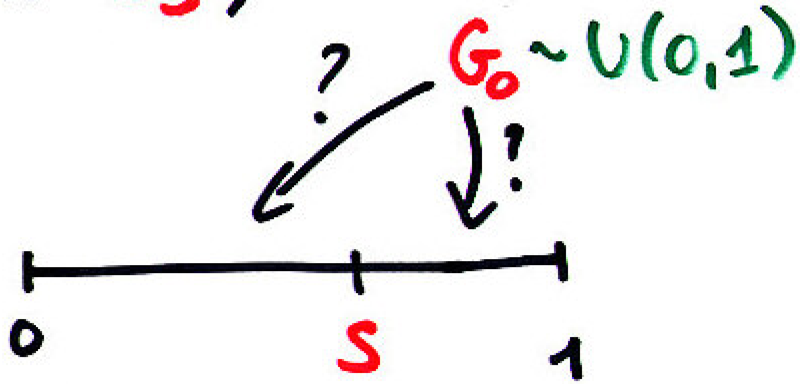
- how to sample an event of probability s?
  (an s-coin C_s)

- Let $s \in [0,1]$ and $G_0 \sim U(0,1)$

- how to sample an event of probability s?
  (an s-coin C_s)

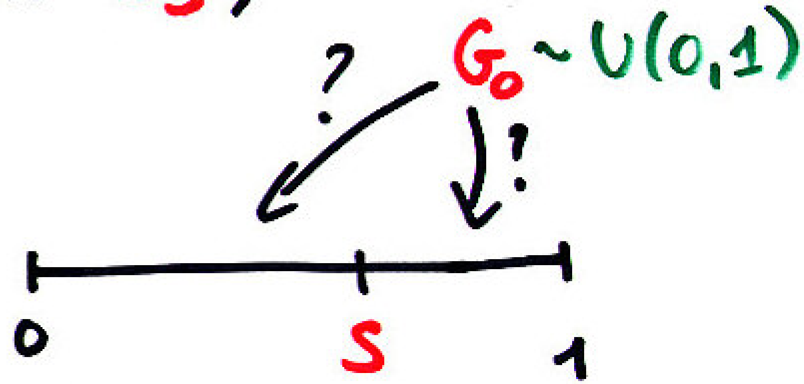
- Let $s \in [0,1]$ and $G_0 \sim U(0,1)$

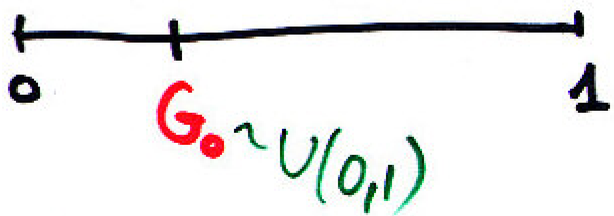
- how to sample an event of probability s?
  (an s-coin C_s)



- Let $s \in [0,1]$ and $G_0 \sim U(0,1)$

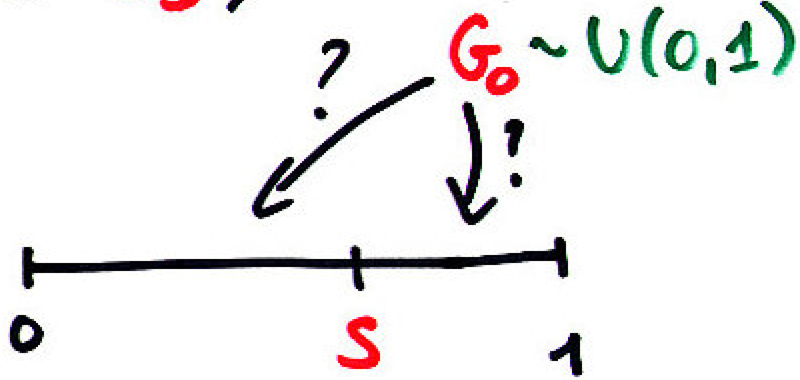
- how to sample an event of probability s?
  (an s-coin C_s)

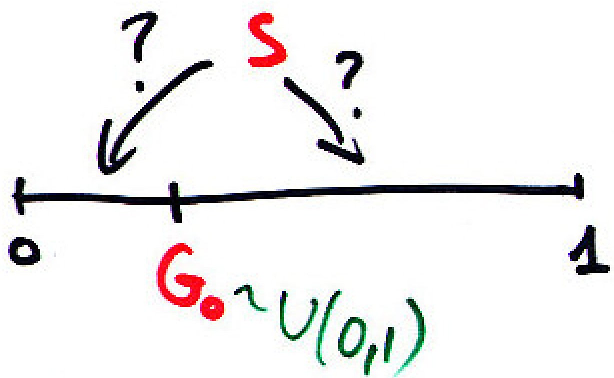




- Let $s \in [0,1]$ and $G_0 \sim U(0,1)$

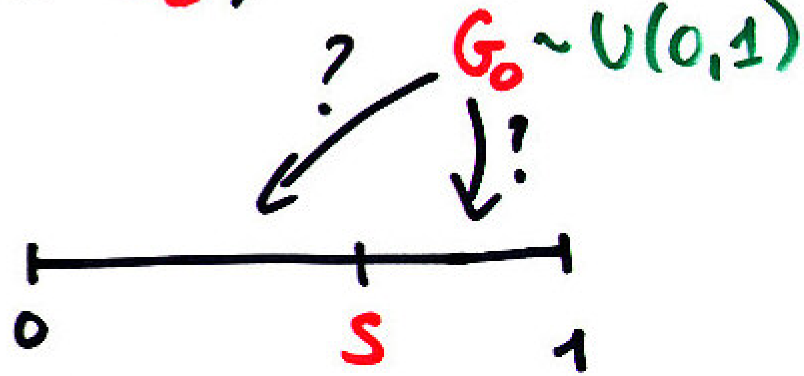
- how to sample an event of probability s?
  (an s-coin C_s)

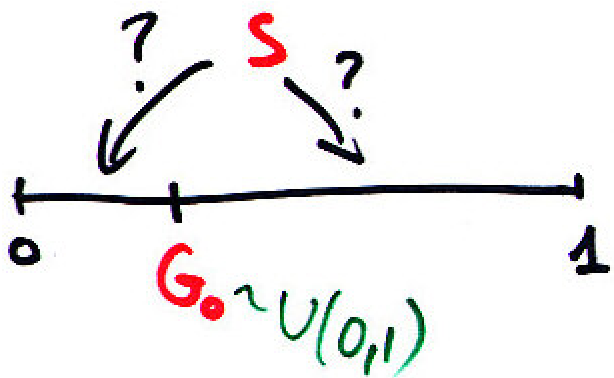




- Let $s \in [0,1]$ and $G_0 \sim U(0,1)$

- how to sample an event of probability s?
  (an s-coin C_s)





- We assume that s is uniquely determined but not known explicitly

- Motivation: often an element of more complex simulation procedures.

- Motivation: often an element of more complex simulation procedures.

- exact simmulation of diffusions
- MCMC inference for diffusion parameter
- rejection sampling
- perfect simulation for Markov chains

- Motivation: often an element of more complex simulation procedures.

- exact simmulation of diffusions
- MCMC inference for diffusion parameter
- rejection sampling
- perfect simulation for Markov chains
- the Bernoulli factory problem

- Motivation: often an element of more complex simulation procedures.

- exact simmulation of diffusions
- MCMC inference for diffusion parameter
- rejection sampling
- perfect simulation for Markov chains
- the Bernoulli factory problem

- Motivation: often an element of more complex simulation procedures.

- exact simmulation of diffusions
- MCMC inference for diffusion parameter
- rejection sampling
- perfect simulation for Markov chains
- the Bernoullie factory problem

how to simulate an $f(p)$-coin given an iid sequence of p-coins

p - UNKNOWN

f KNOWN ex $f(p)=2p$

Assume $l_n \nearrow s$ and $u_n \searrow s$

Assume $l_n \nearrow S$ and $u_n \searrow S$

Algorithm 1

1. $G_0 \sim U(0,1)$; $n = 1$
2. compute l_n, u_n
3. if $G_0 \leq l_n$ set $C_S := 1$
4. if $G_0 \geq u_n$ set $C_S := 0$
5. if $l_n < G_0 < u_n$ set $n = n+1$ and GOTO 2.
6. output C_S

Assume $l_n \nearrow S$ and $u_n \searrow S$

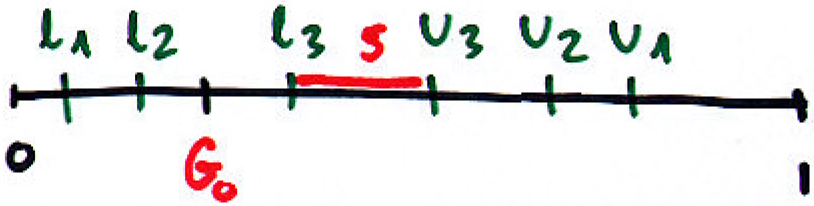
Algorithm 1

1. $G_0 \sim U(0,1)$; $n = 1$
2. compute l_n , u_n
3. if $G_0 \leq l_n$ set $C_S := 1$
4. if $G_0 \geq u_n$ set $C_S := 0$
5. if $l_n < G_0 < u_n$ set $n = n+1$ and GOTO 2.
6. output C_S

Assume $l_n \nearrow s$ and $u_n \searrow s$

Algorithm 1

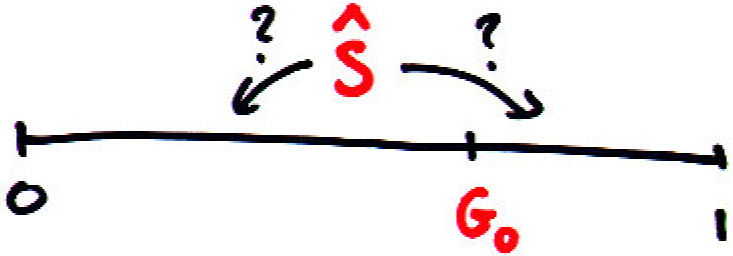
1. $G_0 \sim U(0,1)$; $n = 1$
2. compute l_n, u_n
3. if $G_0 \le l_n$ set $C_s := 1$
4. if $G_0 \ge u_n$ set $C_s := 0$
5. if $l_n < G_0 < u_n$ set $n = n+1$ and GOTO 2.
6. output C_s

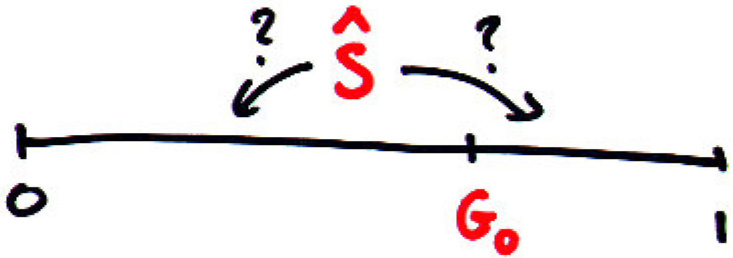


Assume $l_n \nearrow s$ and $u_n \searrow s$

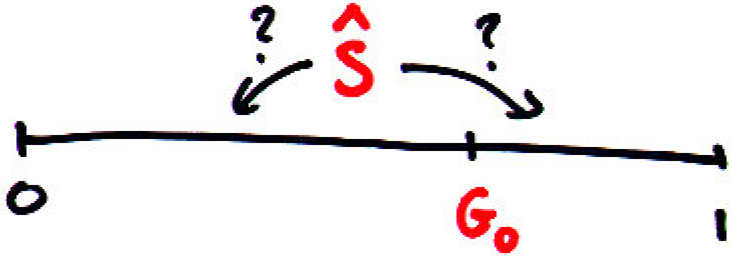
Algorithm 1

1. $G_0 \sim U(0,1)$; $n = 1$
2. compute l_n , u_n
3. if $G_0 \leq l_n$ set $C_s := 1$
4. if $G_0 \geq u_n$ set $C_s := 0$
5. if $l_n < G_0 < u_n$ set $n = n+1$ and GOTO 2.
6. output C_s


?
Ŝ
?
0
G0
1



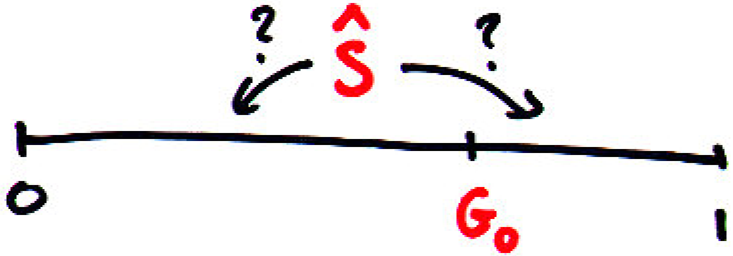
Assume $\hat{S}$ is an unbiased estimator of S.



Assume $\hat{S}$ is an unbiased estimator of S.

Algorithm 2

$$C_S := \mathbb{1}\{G_0 \leq \hat{S}\}$$

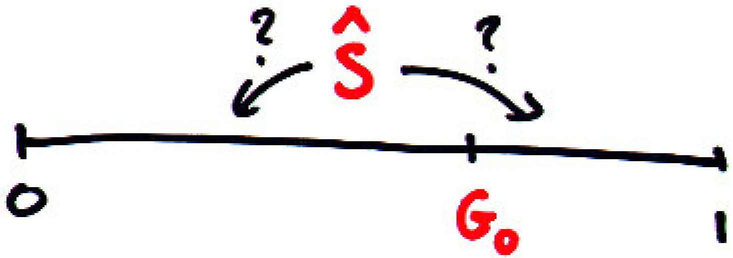


Assume $\hat{S}$ is an unbiased estimator of S.

Algorithm 2

$$C_S := \mathbb{1}\{G_0 \leq \hat{S}\}$$

$$P(C_S = 1) = E\,\mathbb{1}(G_0 \leq \hat{S})$$



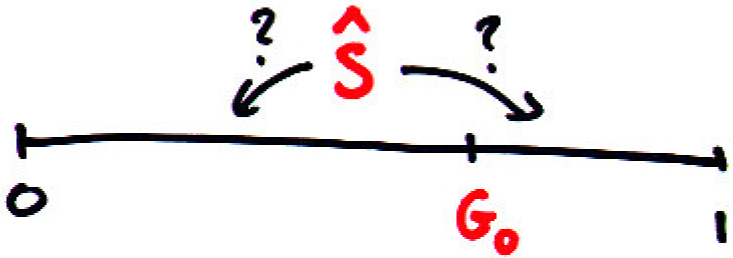
Assume $\hat{S}$ is an unbiased estimator of S.

Algorithm 2

$$C_S := \mathbb{1}\{G_0 \le \hat{S}\}$$

$$P(C_S = 1) = E\,\mathbb{1}(G_0 \le \hat{S})$$

$$= E\left(E\left(\mathbb{1}(G_0 \le \hat{s}) \mid \hat{S} = \hat{s}\right)\right)$$



Assume $\hat{S}$ is an unbiased estimator of S.

Algorithm 2

$$C_S := \mathbb{1}\{G_0 \le \hat{S}\}$$

$$\begin{aligned} P(C_S = 1) &= E\,\mathbb{1}(G_0 \le \hat{S}) \\ &= E\left(E\left(\mathbb{1}(G_0 \le \hat{s}) \,\middle|\, \hat{S} = \hat{s}\right)\right) \\ &= E\hat{S} = S. \end{aligned}$$

L_n and U_n - unbiased estimators of l_n and u_n

L_n and U_n - unbiased estimators of l_n and u_n

- $P(L_n \leq U_n) = 1$
- $P(L_n \in [0,1]) = 1$ and $P(U_n \in [0,1]) = 1$
- $E L_n = l_n \nearrow S$ and $E U_n = u_n \searrow S$

L_n and U_n - unbiased estimators of l_n and u_n (6)

- $P(L_n \leq U_n) = 1$
- $P(L_n \in [0,1]) = 1$ and $P(U_n \in [0,1]) = 1$
- $E\, L_n = l_n \nearrow s$ and $E\, U_n = u_n \searrow s$

(*) $P(L_{n-1} \leq L_n) = 1$ and $P(U_{n-1} \geq U_n) = 1$

L_n and U_n - unbiased estimators of l_n and u_n

- $P(L_n \le U_n) = 1$
- $P(L_n \in [0,1]) = 1$ and $P(U_n \in [0,1]) = 1$
- $E\,L_n = l_n \nearrow s$ and $E\,U_n = u_n \searrow s$

(*) $P(L_{n-1} \le L_n) = 1$ and $P(U_{n-1} \ge U_n) = 1$

Algorithm 3

1. $G_0 \sim U(0,1)$; $n = 1$
2. obtain L_n and U_n given L_{n-1}, U_{n-1}
3. if $G_0 \le L_n$ ...

   $G_0 \ge U_n$ ...

   $L_n < G_0 < U_n$ ...

L_n and U_n - unbiased estimators of l_n and u_n

- $P(L_n \leq U_n) = 1$
- $P(L_n \in [0,1]) = 1$ and $P(U_n \in [0,1]) = 1$
- $E L_n = l_n \nearrow s$ and $E U_n = u_n \searrow s$

(*) $P(L_{n-1} \leq L_n) = 1$ and $P(U_{n-1} \geq U_n) = 1$

Algorithm 3

1. $G_0 \sim U(0,1)$; $n = 1$
2. obtain L_n and U_n given L_{n-1}, U_{n-1}
3. if $G_0 \leq L_n$ ...

   $G_0 \geq U_n$ ...

   $L_n < G_0 < U_n$ ...

Lemma

Algorithm 3 outputs a valid s-coin

L_n and U_n - unbiased estimators of l_n and u_n

- $P(L_n \leq U_n) = 1$
- $P(L_n \in [0,1]) = 1$ and $P(U_n \in [0,1]) = 1$
- $E L_n = l_n \nearrow s$ and $E U_n = u_n \searrow s$

$(*)$ $P(L_{n-1} \leq L_n) = 1$ and $P(U_{n-1} \geq U_n) = 1$

Algorithm 3

1. $G_0 \sim U(0,1)$; $n = 1$
2. obtain L_n and U_n given L_{n-1}, U_{n-1}
3. if $G_0 \leq L_n$ ...

   $G_0 \geq U_n$ ...

   $L_n < G_0 < U_n$ ...

$\mathcal{F}_0 = \{\phi, \Omega\}$ $\mathcal{F}_n = \sigma(L_n, U_n)$

$\mathcal{F}_{k,n} = \sigma(\mathcal{F}_k, \mathcal{F}_{k+1}, \dots \mathcal{F}_n)$

given $\mathcal{F}_{0,n-1}$

Lemma

Algorithm 3 outputs a valid s-coin

(**) L_n - reverse time supermartingale

U_n - reverse time submartingale

(**) L_n - reverse time supermartingale

U_n - reverse time submartingale

$$E(L_{n-1} \mid \mathcal{F}_{n,\infty}) = E(L_{n-1} \mid \mathcal{F}_n) \leq L_n$$

$$E(U_{n-1} \mid \mathcal{F}_{n,\infty}) = E(U_{n-1} \mid \mathcal{F}_n) \geq U_n$$

(**) L_n - reverse time supermartingale
U_n - reverse time submartingale

$$E(L_{n-1} | \mathcal{F}_{n,\infty}) = E(L_{n-1} | \mathcal{F}_n) \leq L_n$$

$$E(U_{n-1} | \mathcal{F}_{n,\infty}) = E(U_{n-1} | \mathcal{F}_n) \geq U_n$$

<u>Algorithm 4</u>

1. $G_0 \sim U(0,1)$; $n=1$; $L_0 \equiv \tilde{L}_0 \equiv 0$ and $U_0 \equiv \tilde{U}_0 \equiv 1$

(**) L_n - reverse time supermartingale
U_n - reverse time submartingale

$$E(L_{n-1} | \mathcal{F}_{n,\infty}) = E(L_{n-1} | \mathcal{F}_n) \leq L_n$$

$$E(U_{n-1} | \mathcal{F}_{n,\infty}) = E(U_{n-1} | \mathcal{F}_n) \geq U_n$$

<u>Algorithm 4</u>

1. $G_0 \sim U(0,1)$; $n = 1$; $L_0 \equiv \tilde{L}_0 \equiv 0$ and $U_0 \equiv \tilde{U}_0 \equiv 1$
2. obtain L_n, U_n given $\mathcal{F}_{0,n-1}$
3. compute $L_n^* = E(L_{n-1} | \mathcal{F}_n)$ and $U_n^* = E(U_{n-1} | \mathcal{F}_n)$

(**) L_n - reverse time supermartingale

U_n - reverse time submartingale

$$E(L_{n-1} \mid \mathcal{F}_{n,\infty}) = E(L_{n-1} \mid \mathcal{F}_n) \leq L_n$$

$$E(U_{n-1} \mid \mathcal{F}_{n,\infty}) = E(U_{n-1} \mid \mathcal{F}_n) \geq U_n$$

Algorithm 4

1. $G_0 \sim U(0,1)$; $n = 1$; $L_0 \equiv \tilde{L}_0 \equiv 0$ and $U_0 \equiv \tilde{U}_0 \equiv 1$
2. obtain L_n, U_n given $\mathcal{F}_{0,n-1}$
3. compute $L_n^* = E(L_{n-1} \mid \mathcal{F}_n)$ and $U_n^* = E(U_{n-1} \mid \mathcal{F}_n)$
4. compute

$$\tilde{L}_n = \tilde{L}_{n-1} + \frac{L_n - L_n^*}{U_n^* - L_n^*} (\tilde{U}_{n-1} - \tilde{L}_{n-1})$$

(**) L_n - reverse time supermartingale
U_n - reverse time submartingale

$$E(L_{n-1} | \mathcal{F}_{n,\infty}) = E(L_{n-1} | \mathcal{F}_n) \leq L_n$$

$$E(U_{n-1} | \mathcal{F}_{n,\infty}) = E(U_{n-1} | \mathcal{F}_n) \geq U_n$$

Algorithm 4

1. $G_0 \sim U(0,1)$; $n = 1$; $L_0 \equiv \tilde{L}_0 \equiv 0$ and $U_0 \equiv \tilde{U}_0 \equiv 1$
2. obtain L_n, U_n given $\mathcal{F}_{0,n-1}$
3. compute $L_n^* = E(L_{n-1} | \mathcal{F}_n)$ and $U_n^* = E(U_{n-1} | \mathcal{F}_n)$
4. compute

$$\tilde{L}_n = \tilde{L}_{n-1} + \frac{L_n - L_n^*}{U_n^* - L_n^*}(\tilde{U}_{n-1} - \tilde{L}_{n-1}) \qquad \tilde{U}_n = \tilde{U}_{n-1} + \frac{U_n^* - U_n}{U_n^* - L_n^*}(\tilde{U}_{n-1} - \tilde{L}_{n-1})$$

(**) L_n - reverse time supermartingale
U_n - reverse time submartingale

$$E(L_{n-1} | \mathcal{F}_{n,\infty}) = E(L_{n-1} | \mathcal{F}_n) \leq L_n$$

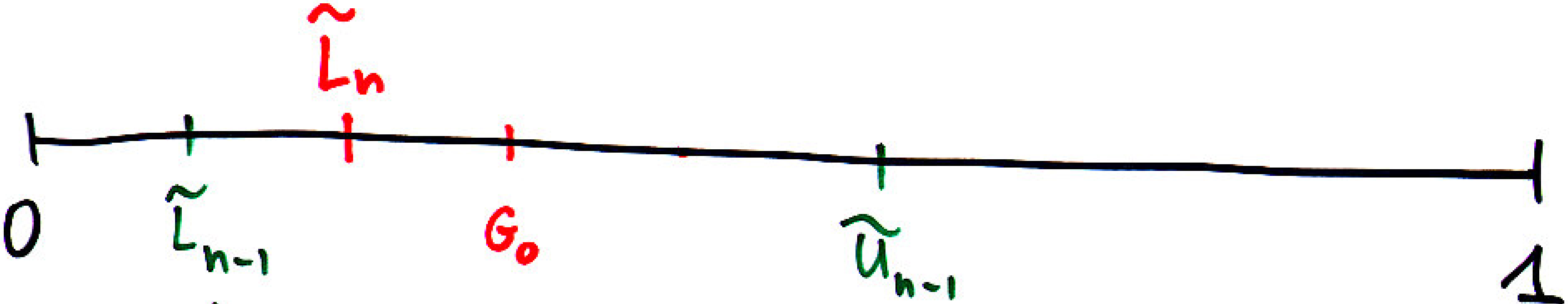
$$E(U_{n-1} | \mathcal{F}_{n,\infty}) = E(U_{n-1} | \mathcal{F}_n) \geq U_n$$

Algorithm 4

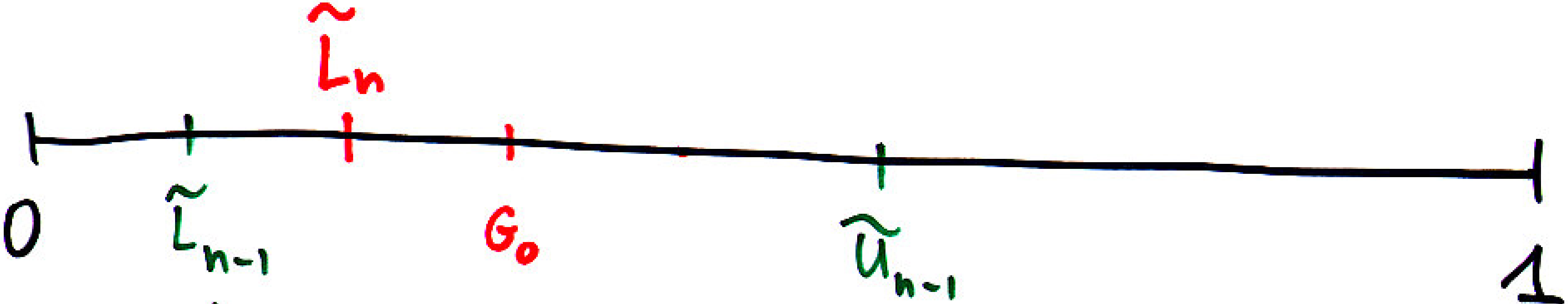
1. $G_0 \sim U(0,1)$; $n = 1$; $L_0 \equiv \tilde{L}_0 \equiv 0$ and $U_0 \equiv \tilde{U}_0 \equiv 1$
2. obtain L_n, U_n given $\mathcal{F}_{0,n-1}$
3. compute $L_n^* = E(L_{n-1} | \mathcal{F}_n)$ and $U_n^* = E(U_{n-1} | \mathcal{F}_n)$
4. compute
$$\tilde{L}_n = \tilde{L}_{n-1} + \frac{L_n - L_n^*}{U_n^* - L_n^*}(\tilde{U}_{n-1} - \tilde{L}_{n-1}) \qquad \tilde{U}_n = \tilde{U}_{n-1} + \frac{U_n^* - U_n}{U_n^* - L_n^*}(\tilde{U}_{n-1} - \tilde{L}_{n-1})$$
5. if $G_0 \leq \tilde{L}_n$ set $C_s := 1$
6. if $G_0 \geq \tilde{U}_n$ set $C_s := 0$

....


0
$\tilde{L}_{n-1}$
G_0
$\tilde{U}_{n-1}$
1

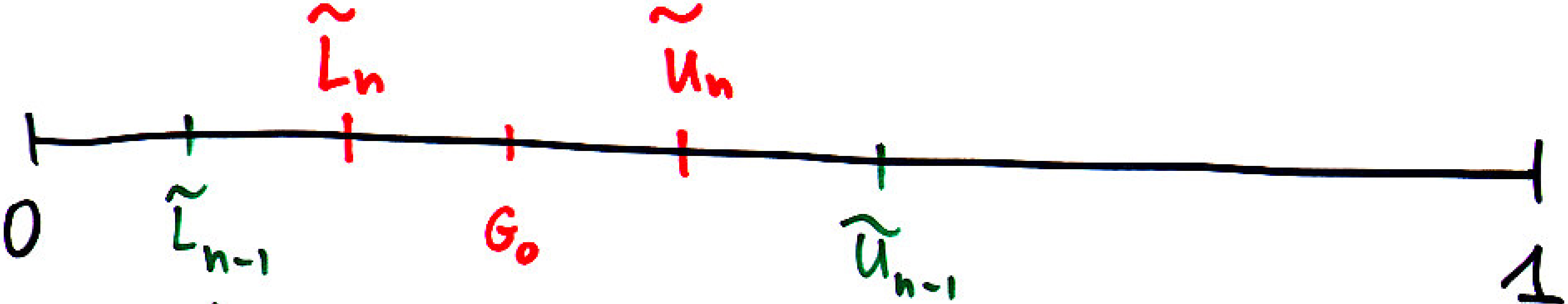


$$\tilde{L}_n = \tilde{L}_{n-1} + \frac{L_n - L_n^*}{U_n^* - L_n^*}\left(\tilde{U}_{n-1} - \tilde{L}_{n-1}\right)$$



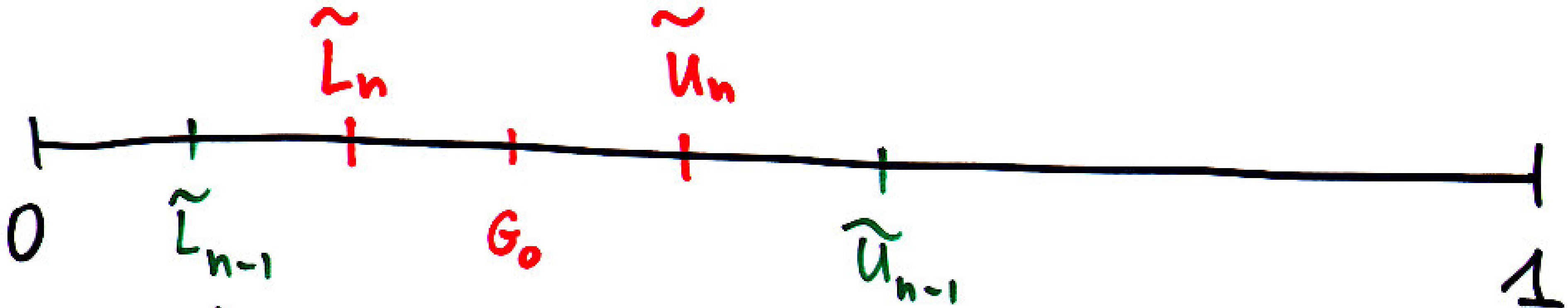
$$\tilde{L}_n = \tilde{L}_{n-1} + \frac{L_n - L_n^*}{U_n^* - L_n^*}\left(\tilde{U}_{n-1} - \tilde{L}_{n-1}\right)$$

$$\tilde{U}_n = \tilde{U}_{n-1} + \frac{U_n^* - U_n}{U_n^* - L_n^*}\left(\tilde{U}_{n-1} - \tilde{L}_{n-1}\right)$$



$$\tilde{L}_n = \tilde{L}_{n-1} + \frac{L_n - L_n^*}{U_n^* - L_n^*}\left(\tilde{U}_{n-1} - \tilde{L}_{n-1}\right)$$

$$\tilde{U}_n = \tilde{U}_{n-1} + \frac{U_n^* - U_n}{U_n^* - L_n^*}\left(\tilde{U}_{n-1} - \tilde{L}_{n-1}\right)$$



$$\tilde{L}_n = \tilde{L}_{n-1} + \frac{L_n - L_n^*}{U_n^* - L_n^*}\left(\tilde{U}_{n-1} - \tilde{L}_{n-1}\right)$$

$$\tilde{U}_n = \tilde{U}_{n-1} + \frac{U_n^* - U_n}{U_n^* - L_n^*}\left(\tilde{U}_{n-1} - \tilde{L}_{n-1}\right)$$

This construction preserves expectation and $\tilde{L}_n, \tilde{U}_n$ are monotone.

$$\tilde{L}_n = \tilde{L}_{n-1} + \frac{L_n - L_n^*}{U_n^* - L_n^*}\left(\tilde{U}_{n-1} - \tilde{L}_{n-1}\right)$$

$$\tilde{U}_n = \tilde{U}_{n-1} + \frac{U_n^* - U_n}{U_n^* - L_n^*}\left(\tilde{U}_{n-1} - \tilde{L}_{n-1}\right)$$

This construction preserves expectation and $\tilde{L}_n, \tilde{U}_n$ are monotone.

__Thm.__ Algorithm 4 outputs a valid s-coin.

The Bernoulli Factory Problem

9

input: $X_1, X_2, \ldots$ sequence of p-coins (p is UNKNOWN)
$p \in (0, 1/2)$

output: Y a single $2p$-coin

The Bernoulli Factory Problem

input: $X_1, X_2, \ldots$ sequence of p-coins (p is UNKNOWN)
$p \in (0, 1/2)$

output: Y a single $2p$-coin $\rightarrow$ IMPOSSIBLE!

The Bernoulli Factory Problem

input: $X_1, X_2, \ldots$ sequence of p-coins (p is UNKNOWN)
$p \in (0, 1/2)$

output: Y a single $2p$-coin $\rightarrow$ IMPOSSIBLE!

Y a single $f(p) = \min\{2p, 1-2\varepsilon\}$-coin

The Bernoulli Factory Problem

input: $X_1, X_2, \ldots$ sequence of p-coins (p is UNKNOWN)
$p \in (0, 1/2)$

output: Y a single $2p$-coin $\rightarrow$ IMPOSSIBLE!

Y a single $f(p) = \min\{2p, 1-2\varepsilon\}$-coin

Proposition: An algorithm that simulates f on $\mathcal{P} \subseteq (0,1)$ exists if and only if for all $n \geq 1$ there exist

The Bernoulli Factory Problem

input: $X_1, X_2, \ldots$ sequence of p-coins (p is UNKNOWN)
$p \in (0, 1/2)$

output: Y a single $2p$-coin $\rightarrow$ IMPOSSIBLE!

Y a single $f(p) = \min\{2p, 1-2\varepsilon\}$-coin

Proposition: An algorithm that simulates f on $\mathcal{P} \subseteq (0,1)$ exists if and only if for all $n \geq 1$ there exist polynomials $g_n(p)$ and $h_n(p)$ of the form

$$g_n(p) = \sum_{k=0}^{n} \binom{n}{k} a(n,k) p^k (1-p)^{n-k} \qquad h_n(p) = \sum_{k=0}^{n} \binom{n}{k} b(n,k) p^k (1-p)^{n-k}$$

The Bernoull Factory Problem

input: $X_1, X_2, \ldots$ sequence of p-coins (p is UNKNOWN)
$p \in (0, 1/2)$

output: Y a single $2p$-coin $\rightarrow$ IMPOSSIBLE!

Y a single $f(p) = \min\{2p, 1-2\varepsilon\}$-coin

Proposition: An algorithm that simulates f on $\mathcal{P} \subseteq (0,1)$ exists if and only if for all $n \geq 1$ there exist polynomials $g_n(p)$ and $h_n(p)$ of the form

$$g_n(p) = \sum_{k=0}^{n} \binom{n}{k} a(n,k) p^k (1-p)^{n-k} \qquad h_n(p) = \sum_{k=0}^{n} \binom{n}{k} b(n,k) p^k (1-p)^{n-k}$$

(i) $0 \leq a(n,k) \leq b(n,k) \leq 1$

The Bernoulli Factory Problem

input: $X_1, X_2, \dots$ sequence of p-coins (p is UNKNOWN)
$p \in (0, 1/2)$

output: Y a single 2p-coin $\rightarrow$ IMPOSSIBLE!

Y a single $f(p) = \min\{2p, 1-2\varepsilon\}$-coin

Proposition: An algorithm that simulates f on $\mathcal{P} \subseteq (0,1)$ exists if and only if for all $n \geq 1$ there exist polynomials $g_n(p)$ and $h_n(p)$ of the form

$$g_n(p) = \sum_{k=0}^{n} \binom{n}{k} a(n,k) p^k (1-p)^{n-k} \qquad h_n(p) = \sum_{k=0}^{n} \binom{n}{k} b(n,k) p^k (1-p)^{n-k}$$

(i) $0 \leq a(n,k) \leq b(n,k) \leq 1$

(ii) $\lim\limits_{n \to \infty} g_n(p) = f(p) = \lim\limits_{n \to \infty} h_n(p)$

The Bernoulli Factory Problem

input: $X_1, X_2, \dots$ sequence of p-coins (p is UNKNOWN)
$p \in (0, 1/2)$

output: Y a single $2p$-coin $\to$ IMPOSSIBLE!

Y a single $f(p) = \min\{2p, 1-2\varepsilon\}$-coin

Proposition: An algorithm that simulates f on $\mathcal{P} \subseteq (0,1)$ exists if and only if for all $n \geq 1$ there exist polynomials $g_n(p)$ and $h_n(p)$ of the form

$$g_n(p) = \sum_{k=0}^{n} \binom{n}{k} a(n,k) p^k (1-p)^{n-k} \qquad h_n(p) = \sum_{k=0}^{n} \binom{n}{k} b(n,k) p^k (1-p)^{n-k}$$

(i) $0 \leq a(n,k) \leq b(n,k) \leq 1$

(ii) $\lim_{n\to\infty} g_n(p) = f(p) = \lim_{n\to\infty} h_n(p)$

(iii) for all $m < n$ coefficients satisfy

$$a(n,k) \geq \sum_{i=0}^{k} \frac{\binom{n-m}{k-i}\binom{m}{i}}{\binom{n}{k}} a(m,i) \qquad b(n,k) \leq \sum_{i=0}^{k} \frac{\binom{n-m}{k-i}\binom{m}{i}}{\binom{n}{k}} b(m,i)$$

polynomials $\Rightarrow$ algorithm

$X_1, X_2, \ldots$ p-coins if $\sum_{i=1}^{n} X_i = k$, let $L_n = a(n,k)$

$U_n = b(n,k)$

The rest of the proof:

check that assumptions for L_n and U_n

in ALgorithm 4 hold!