Fast Simulation Of New Coins From Old

Şerban Nacu* Yuval Peres†

December 22, 2003

Abstract

Let $S \subset (0,1)$. Given a known function $f : S \to (0,1)$, we consider the problem of using independent tosses of a coin with probability of heads p (where $p \in S$ is unknown) to simulate a coin with probability of heads $f(p)$. We prove that if S is a closed interval and f is real analytic on S, then f has a fast simulation on S (the number of p-coin tosses needed has exponential tails). Conversely, if a function f has a fast simulation on an open set, then it is real analytic on that set.

1 Introduction

We consider the problem of using a coin with probability of heads p (p unknown) to simulate a coin with probability of heads $f(p)$, where f is some known function. By this we mean the following: we are allowed to toss the original p-coin as many times as we want. We stop at some (almost surely) finite stopping time N, and depending on the outcomes of the first N tosses, we declare heads or tails. We want the probability of declaring a head to be exactly $f(p)$.

This problem goes back to Von Neumann's 1951 article [11], where he describes an algorithm which simulates the constant function $f(p) \equiv 1/2$. It is natural to ask whether this is possible for other functions, and in 1991 S. Asmussen raised the question for the function $f(p) = 2p$, where it is known that $p \in (0, 1/2)$ (see [8]). The same question was raised independently but later by J. Propp (see [10]).

In 1994, Keane and O'Brien [8] obtained a necessary and sufficient condition for such a simulation to be possible. Consider $f : S \to [0,1]$, where $S \subset (0,1)$. Then it is possible to simulate a coin with probability of heads $f(p)$ for all $p \in S$ if and only if f is constant, or f is continuous and satisfies, for some $n \geq 1$,

$$\min(f(p), 1-f(p)) \geq \min(p, 1-p)^n \quad \forall p \in S. \tag{1}$$

*Department of Statistics, UC Berkeley, CA 94720, USA. serban@stat.berkeley.edu. http://stat-www.berkeley.edu/~serban

†Departments of Statistics and Mathematics, UC Berkeley, CA 94720, USA. peres@stat.berkeley.edu. http://stat-www.berkeley.edu/~peres. Research supported in part by NSF Grants DMS-0104073 and DMS-0244479.

In particular, $f(p) = 2p$ cannot be simulated on $(0, 1/2)$, since the inequality (1) cannot hold for p close to $1/2$. However, if we are given $\epsilon > 0$, then an algorithm exists to simulate a $2p$-coin from tosses of a p-coin for $p \in (0, 1/2 - \epsilon)$.

The methods in [8] do not provide any estimates on the number N of p-coin tosses needed to simulate an $f(p)$-coin. The stopping time N will typically be unbounded, and for fast algorithms it should have rapidly decaying tails. For example, in Von Neumann's algorithm [11], the tail probabilities satisfy $\mathbf{P}_p(N > n) \leq (p^2 + (1-p)^2)^{\lfloor n/2 \rfloor}$, so they decay exponentially in n.

Definition 1. *A function f has a* **fast simulation** *on S if there exists an algorithm which simulates f on S, and for any $p \in S$ there exist constants $C > 0, \rho < 1$ (which may depend on p) such that the number N of required inputs satisfies $\mathbf{P}_p(N > n) \leq C\rho^n$.*

Remark. If S is closed and f has a fast simulation on S, then we can choose constants C, ρ not depending on $p \in S$. See Proposition 21 for a proof.

Theorem 1. *For any $\epsilon > 0$, the function $f(p) = 2p$ has a fast simulation on $[0, 1/2 - \epsilon]$.*

Building on this result, we prove

Theorem 2. *If $f : I \to (0, 1)$ is real analytic on the closed interval $I \subset (0, 1)$, then it has a fast simulation on I. Conversely, if a function has a fast simulation, then it is real analytic on any open subset of its domain.*

As the results stated above indicate, there is a correspondence between properties of simulation algorithms and classes of functions. The following table summarizes the results of [8], [10] and the present paper on this correspondence. For simplicity, in this table we restrict attention to functions $f : S \mapsto T$ where S, T are closed intervals in $(0, 1)$.

Simulation type		Function class	Reference
Terminating a.s.	$\Leftrightarrow$	f continuous	[8]
With finite expectation	$\Rightarrow$	f Lipshitz	Proposition 23
With finite k'th moment (and uniform tails)	$\Rightarrow$	$f \in C^k$	Proposition 22
Fast (with exponential tails)	$\Leftrightarrow$	f real analytic	Theorem 2
Via pushdown automaton	$\Rightarrow$	f algebraic over $\mathbf{Q}$	[10]
Via finite automaton	$\Leftrightarrow$	f rational over $\mathbf{Q}$ and $f((0,1)) \subset (0,1)$	[10]

We do not know whether the one-sided arrows above can be reversed.

We prove Theorem 1 in Sections 2 and 3. In Section 2 we show that simulating f is equivalent to finding sequences of certain Bernstein polynomials which approximate f from above and below. If the approximations are good, then the simulations are fast. In Section 3 we use this to construct a fast simulation for the function $2p$. We can do this because the Bernstein polynomials provide exponentially convergent approximations for linear functions.

In Section 4 we prove the sufficient (constructive) part of Theorem 2. This is done in several steps. First, once we have a fast simulation for $2p$, it is easy to construct fast simulations for polynomials. Using an auxiliary geometric random variable, we also obtain fast simulations for functions which have a series expansion around the origin. This proves Theorem 2 for real analytic functions that extend to an analytic function on a disk centered at the origin. For a general real analytic function, we use Möbius maps of the form $(az+b)/(cz+d)$ to map a subset of their domain to the unit disk. Since we have fast simulations for Möbius maps, this leads to fast simulations for the original function.

In particular, Theorem 2 guarantees fast simulations for any rational function f, over any subset of $(0,1)$ where $\epsilon \leq f \leq 1-\epsilon$. This generalizes a result from [10], where the authors prove that any rational function $f:(0,1)\to(0,1)$ has a simulation by a finite automaton, which is fast.

In Section 5 we prove the necessary part of Theorem 2, and in Section 6 we describe a very simple algorithm that gives a good approximate simulation for the function $2p$ (the error decreases exponentially in the number of steps). In Section 7 we give a simple proof of the fact that any continuous function bounded away from 0 and 1 has a simulation. Finally, in Section 8 we mention some open problems.

2 Simulation as an Approximation Problem

In this section we show that a function f can be simulated if and only if it can be approximated by certain polynomials, both from below and from above, and the approximations converge to f. Furthermore, the speed of convergence of the approximations determines the speed of the simulation (i.e., the distribution of the number of coin tosses needed).

Let $\mathbf{P}_p$ be the law of an infinite sequence $\mathbf{X} = (X_1, X_2, \ldots)$ of i.i.d. coin tosses with probability of heads p. By a slight abuse of notation, we also denote by $\mathbf{P}_p$ the induced law of the first n tosses $X_1, \ldots, X_n$, so for $A \subset \{0,1\}^n$, $\mathbf{P}_p(A) = \mathbf{P}_p((X_1, \ldots, X_n) \in A)$.

Fix n and consider the first n tosses. Either the algorithm terminates after at most n inputs (and in that case, it outputs a 1 or a 0), or it needs more than n inputs. Let $A_n \subset \{0,1\}^n$ be the set of inputs where the algorithm terminates and outputs 1, and let B_n be the set of inputs where either the algorithm terminates and outputs 1, or needs more than n inputs. Then clearly

$$\mathbf{P}_p(A_n) \leq \mathbf{P}_p(\text{algorithm outputs 1}) \leq \mathbf{P}_p(B_n).$$

The middle term is $f(p)$. Any sequence in $\{0,1\}^n$ has probability $p^k(1-p)^{n-k}$, where k is the number of 1's in the sequence, so the lower and upper bounds are polynomials of the form $\sum_k c_k p^k (1-p)^{n-k}$, with c_k non-negative integers. The probability that the algorithm needs more than n inputs is $\mathbf{P}_p(B_n) - \mathbf{P}_p(A_n)$, so if the polynomials are good approximations for f, then the number of inputs needed has small tails.

It is less obvious that a converse also holds: given a function f and a sequence of approximating polynomials with certain properties, there exists an algorithm which generates f, so that the probabilities of A_n and B_n as defined above are given by the approximating polynomials. We prove this in the rest of this section.

In order to state our result in a compact form, we introduce the following

Definition 2. *Let $q(x,y), r(x,y)$ be homogenous polynomials of equal degree with real coefficients. If all coefficients of $r-q$ are non-negative, then we write $q \preceq r$. If in addition $q \neq r$, then we write $q \prec r$.*

This defines a partial order on the set of homogenous polynomials of two variables. If $q \preceq r$, then clearly $q(x,y) \leq r(x,y)$ for all $x, y \geq 0$. The converse does not hold; for example, $xy \leq x^2+y^2$ for all $x,y \geq 0$, but $xy \not\preceq x^2+y^2$.

Proposition 3. *If there exists an algorithm which simulates a function f on a set $S \subset (0,1)$, then for all $n \geq 1$ there exist polynomials*

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

with the following properties:

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

(ii) $\binom{n}{k} a(n,k)$ *and* $\binom{n}{k} b(n,k)$ *are integers.*

(iii) $\lim_n g_n(p, 1-p) = f(p) = \lim_n h_n(p,1-p)$ *for all* $p \in S$.

(iv) *For all $m < n$, we have $(x+y)^{n-m} g_m(x,y) \preceq g_n(x,y)$ and $h_n(x,y) \preceq (x+y)^{n-m} h_m(x,y)$.*

Conversely, if there exist such polynomials $g_n(x,y), h_n(x,y)$ satisfying (i)-(iv), then there exists an algorithm which simulates f on S, such that the number N of inputs needed satisfies $\mathbf{P}_p(N > n) = h_n(p,1-p) - g_n(p,1-p)$.

Proof. $\Rightarrow$ Suppose an algorithm exists, consider its first n inputs, and define as above $A_n \subset \{0,1\}^n$ to be the set of inputs where the algorithm outputs 1, and $B_n \subset \{0,1\}^n$ the set where the algorithm outputs 1 or needs more than n inputs. We also partition $A_n = \bigcup A_{n,k}$ and $B_n = \bigcup B_{n,k}$ according to the number k of 1's in each word. Then every element in $A_{n,k}$ or $B_{n,k}$ has probability $p^k(1-p)^{n-k}$, so if we define

$$a(n,k) = |A_{n,k}| / \binom{n}{k}, \quad b(n,k) = |B_{n,k}| / \binom{n}{k}$$

then

$$g_n(p, 1-p) = \mathbf{P}_p(A_n), \quad h_n(p,1-p) = \mathbf{P}_p(B_n).$$

Condition (i) and (ii) are clearly satisfied, and (iii) also follows easily. As discussed above, we have $g_n(p, 1-p) \leq f(p) \leq h_n(p, 1-p)$ and $\mathbf{P}_p(N > n) = h_n(p, 1-p) - g_n(p, 1-p)$; since the algorithm terminates almost surely, the difference must converge to 0. From the definition of A_n and B_n, it is clear that $g_n(p, 1-p)$ is an increasing sequence, and $h_n(p, 1-p)$ is decreasing.

Condition (iv) must hold because of the structure of the sets A_n and B_n. Indeed, let $m < n$ and assume $(X_1, \ldots, X_m) \in A_m$. Then $(X_1, \ldots, X_n) \in A_n$, whatever values $X_{m+1}, \ldots, X_n$ take. To make this formal, for $E \subset \{0,1\}^m$ define

$$T_{m,n}(E) = \{(X_1, \ldots, X_n) \in \{0,1\}^n : (X_1, \ldots, X_m) \in E\}.$$

That is, $T_{m,n}(E)$ is the set obtained by taking each element in E and adding at the end all possible combinations of $n-m$ zeroes and ones. Partition $T_{m,n}(E) = \bigcup T^k_{m,n}(E)$, so that all words in $T^k_{m,n}(E)$ have exactly k 1's. We have $T_{m,n}(A_m) \subset A_n$, so $T^k_{m,n}(A_m) \subset A_{n,k}$, so

$$|A_{n,k}| \geq |T^k_{m,n}(A_m)| = \sum_{i=0}^{k} \binom{n-m}{k-i} |A_{m,i}|.$$

which is the same as

$$\binom{n}{k} a(n,k) \geq \sum_{i=0}^{k} \binom{n-m}{k-i} \binom{m}{i} a(m,i)\,; \tag{2}$$

this is equivalent to $g_n(x,y) \succeq (x+y)^m g_m(x,y)$. A similar observation holds for the sets B_n, and this completes the proof of (iv).

$\Leftarrow$ Given the numbers $a(n,k), b(n,k)$ satisfying (i)-(iv), we shall define inductively sets $A_n = \bigcup A_{n,k}$, $B_n = \bigcup B_{n,k}$ with

$$A_{n,k} \subset B_{n,k}, \quad |A_{n,k}| = \binom{n}{k} a(n,k), \quad |B_{n,k}| = \binom{n}{k} b(n,k).$$

We also want the extra property that if $m < n$ then $T_{m,n}(A_m) \subset A_n$ and $T_{m,n}(B_m) \supset B_n$. Then we can construct an algorithm simulating f as follows: at step n, output 1 if in A_n, output 0 if in B_n^c, continue if in $B_n - A_n$.

We define $A_{1,0} = \{0\}$ if $a(1,0) = 1$, and $\emptyset$ otherwise. We define $A_{1,1} = \{1\}$ if $a(1,1) = 1$, and $\emptyset$ otherwise. Similarly for $B_{1,0}$ and $B_{1,1}$. Since $a(1,k) \leq b(1,k)$, we have $A_{1,k} \subset B_{1,k}$ for $k = 0, 1$. Condition (iv) guarantees that if

$$|A_{m,k}| = \binom{m}{k} a(m,k) \text{ and } |B_{m,k}| = \binom{m}{k} b(m,k)$$

for all k, then

$$|T^k_{m,n}(A_m)| \leq \binom{n}{k} a(n,k) \leq \binom{n}{k} b(n,k) \leq |T^k_{m,n}(B_m)|. \tag{3}$$

Hence we can construct the sets A_n, B_n from the sets A_m, B_m as follows. We want to have

$$T^k_{m,n}(A_m) \subset A_{n,k} \subset B_{n,k} \subset T^k_{m,n}(B_m). \tag{4}$$

In view of (3), this can be done by simply choosing any total ordering of the set of binary words of length n with k ones. We build $A_{n,k}$ by starting with $T^k_{m,n}(A_m)$ and then adding elements of $T^k_{m,n}(B_m)$ in increasing order until we obtain the desired cardinality $\binom{n}{k}a(n,k)$. Then we add $\binom{n}{k}b(n,k) - \binom{n}{k}a(n,k)$ extra elements to obtain $B_{n,k}$. Of course, $A_n = \bigcup A_{n,k}$ and $B_n = \bigcup B_{n,k}$. It is immediate that the sets thus defined have the desired properties, so the induction step from m to $n = m+1$ works and the proof is complete. □

Remark A. Condition (iv) in Proposition 3 implies that the sequence $(g_n(p, 1-p))_{n\geq 1}$ is increasing, and the sequence $(h_n(p, 1-p))_{n\geq 1}$ is decreasing (just set $x = p, y = 1-p$).

Remark B. It is enough to define the numbers $a(n,k)$ and $b(n,k)$ when n takes values along an increasing subsequence $n_i \uparrow \infty$. Indeed, assume (iv) holds for $m = n_i, n = n_{i+1}$. Then just like above, we can construct the sets A_n, B_n from the sets A_m, B_m so that (4) holds. Thus we can construct inductively the sets A_{n_i}, B_{n_i}. The algorithm is allowed to stop only at some n_i; if $n_i < n < n_{i+1}$, it just continues. This amounts to defining $A_n = T_{n_i,n}(A_{n_i}), B_n = T_{n_i,n}(B_{n_i})$ for $n_i < n < n_{i+1}$. In terms of the polynomials, this means

$$g_n(x,y) = (x+y)^{n-n_i} g_{n_i}(x,y), \quad h_n(x,y) = (x+y)^{n-n_i} h_{n_i}(x,y)$$

for $n_i < n < n_{i+1}$. This is the same as

$$\begin{aligned} a(n,k) &= (k/n)a(n-1,k-1) + (1-k/n)a(n-1,k)\,, \\ b(n,k) &= (k/n)b(n-1,k-1) + (1-k/n)b(n-1,k) \end{aligned}$$

for $n_i < n < n_{i+1}$ and all $0 \leq k \leq n$. In the next section we will use this for the subsequence of powers of two, $n_i = 2^i$. Note that it is enough to check (iv) for $m = n_i, n = n_{i+1}$, because then the algorithm is well-defined and (iv) must hold for all m, n. Similarly, it is enough to check (iii) for $n = n_i$, because the sequences $(g_n(p, 1-p))_{n\geq 1}$ and $(h_n(p, 1-p))_{n\geq 1}$ are monotone.

Remark C. Finally, condition (ii) in Proposition 3 is not essential. Indeed, suppose we find numbers $\alpha(n,k)$ and $\beta(n,k)$ satisfying all conditions in the proposition, except for (ii). Then if we define

$$a(n,k) = \lfloor \alpha(n,k)\binom{n}{k} \rfloor / \binom{n}{k}, \quad b(n,k) = \lceil \beta(n,k)\binom{n}{k} \rceil / \binom{n}{k} \tag{5}$$

conditions (i) and (ii) are trivially satisfied, and (iv) is satisfied because, for arbitrary x_i non-negative reals and c_i non-negative integers,

$$\lfloor \sum c_i x_i \rfloor \geq \sum c_i \lfloor x_i \rfloor, \quad \lceil \sum c_i x_i \rceil \leq \sum c_i \lceil x_i \rceil. \tag{6}$$

Finally, (iii) still holds for $p \neq 0, 1$ because the error introduced in g_n and h_n is at most $\sum_{k=0}^{n} 2p^k(1-p)^{n-k}$ which is exponentially small.

3 Simulating Linear Functions

Let $\epsilon > 0$, and $f(p) = (2p) \wedge (1-2\epsilon)$. Since we are only interested in small ϵ, we also assume $\epsilon < 1/8$. We will use Proposition 3 to construct an algorithm which simulates f. As explained in Remark B of the previous section, it is enough to define $a(n,k)$ and $b(n,k)$ when n is a power of two. Then the compatibility equations in (iv) are equivalent to

$$a(2n,k)\binom{2n}{k} \geq \sum_{i=0}^{k} a(n,i)\binom{n}{i}\binom{n}{k-i} \tag{7}$$

$$b(2n,k)\binom{2n}{k} \leq \sum_{i=0}^{k} b(n,i)\binom{n}{i}\binom{n}{k-i}. \tag{8}$$

These can be nicely expressed in terms of the hypergeometric distribution.

Definition 3. *We say a random variable X has hypergeometric distribution $H(2n,k,n)$ if*

$$\mathbf{P}(X=i) = \binom{n}{i}\binom{n}{k-i} / \binom{2n}{k} \tag{9}$$

We require $0 \leq k \leq 2n$. If we have an urn with $2n$ balls of which k are red, and we select a sample of n balls uniformly without replacement, then X is the number of red balls in the sample.

In terms of the hypergeometric, the compatibility equations (7), (8) become

$$a(2n,k) \quad \geq \quad \mathbf{E}a(n,X) \tag{10}$$

$$b(2n,k) \quad \leq \quad \mathbf{E}b(n,X). \tag{11}$$

We will need some properties of this distribution:

Lemma 4. *If X has distribution $H(2n,k,n)$ then*

(i) $\mathbf{E}(X/n) = k/2n$

(ii) $\mathbf{Var}(X/n) = k(2n-k)/(4(2n-1)n^2) \leq 1/(2n)$

(iii) *If $a > 0$, then* $\mathbf{P}(|X/n - k/2n| > a) \leq 2\exp(-2a^2 n)$.

Both (i) and (ii) are standard facts; (iii) is a standard large deviation estimate. For a proof, see, for example, [7].

Finally, we need a way to find good approximations for f. Proposition 3, (iii) suggests we can use the Bernstein polynomials. We recall their definition and main property. See [13], chapter 1.4, for more details.

Definition 4. *For any function $f : [0,1] \to \mathbf{R}$ and any integer $n > 0$, the n-th Bernstein polynomial of f is $Q_n(x) = \sum_{k=0}^{n} f(k/n)\binom{n}{k}x^k(1-x)^{n-k}$.*

Proposition 5. *If f is continuous, then $Q_n(x) \to f(x)$ uniformly on $[0,1]$.*

If a function is linear on some interval, the Bernstein polynomials provide a very good approximation to it; this suggests we could use them to construct a fast algorithm for functions such as $f(p) = (2p) \wedge (1-2\epsilon)$. To prove that the compatibility equations (10), (11) hold, we will need the following

Lemma 6. *Let X be hypergeometric with distribution $H(2n,k,n)$ as defined in (9), and let $f : [0,1] \to \mathbf{R}$ be any function with $|f| \le 1$. Then*

(i) *If f is Lipschitz, with $|f(x) - f(y)| \le C|x-y|$, then $|\mathbf{E}f(X/n) - f(k/2n)| \le C/\sqrt{2n}$.*

(ii) *If f is twice differentiable, with $|f''| \le C$, then $|\mathbf{E}f(X/n) - f(k/2n)| \le C/(4n)$.*

(iii) *If f is linear on a neighborhood of $k/2n$, so $f(t) = Ct+D$ if $|t-k/2n| \le a$, then $|\mathbf{E}f(X/n) - f(k/2n)| \le (2|C|+4)\exp(-2a^2n)$.*

Proof. If (i) holds, then we get

$$\begin{aligned} |\mathbf{E}f(X/n) - f(k/2n)| &\le \mathbf{E}|f(X/n) - f(k/2n)| \\ &\le C\mathbf{E}|X/n - k/2n| \\ &\le C(\mathbf{E}|X/n - k/2n|^2)^{1/2} \\ &= C\mathbf{Var}(X/n)^{1/2} \le C/\sqrt{2n}. \end{aligned}$$

If (ii) holds, then Taylor expansion for f gives

$$|f(X/n) - f(k/2n) - (X/n - k/2n)f'(k/2n)| \le (1/2)(X/n - k/2n)^2 \sup|f''|$$

and $\mathbf{E}(X/n - k/2n)f'(k/2n) = 0$, so

$$\begin{aligned} |\mathbf{E}f(X/n) - f(k/2n)| &= |\mathbf{E}(f(X/n) - f(k/2n) - (X/n - k/2n)f'(k/2n))| \\ &\le (C/2)\mathbf{E}(X/n - k/2n)^2 \\ &= (C/2)\mathbf{Var}(X/n) \le C/(4n). \end{aligned}$$

If (iii) holds, then let $g(t) = f(t) - Ct - D$. We have $g = 0$ on $[k/2n - a, k/2n + a]$ and $|g(t) - g(s)| \le |f(t) - f(s)| + |C||t-s| \le 2 + |C|\ \forall t, s \in [0,1]$. Hence

$$\begin{aligned} |\mathbf{E}f(X/n) - f(k/2n)| &= |\mathbf{E}g(X/n) - g(k/2n)| \\ &\le \mathbf{E}|g(X/n) - g(k/2n)| \\ &= \mathbf{E}|g(X/n) - g(k/2n)|1_{|X/n - k/2n| > a} \\ &\le (2+|C|)\mathbf{P}(|X/n - k/2n| > a) \\ &\le 2(2+|C|)\exp(-2a^2n). \end{aligned}$$

This completes the proof of the lemma. □

If we specialize the lemma to $f(p) = (2p) \wedge (1-2\epsilon)$, which is Lipschitz with $C = 2$ and also piecewise linear, we obtain

Proposition 7. *Let $f(p) = (2p) \wedge (1-2\epsilon)$, where $\epsilon < 1/2$. For X satisfying (9), we have*

(i) $|\mathbf{E}f(X/n) - f(k/2n)| \leq \sqrt{2}/\sqrt{n}\, \forall k, n$

(ii) $|\mathbf{E}f(X/n) - f(k/2n)| \leq 8\exp(-2\epsilon^2 n)$ *if* $k/2n \leq 1/2 - 2\epsilon$.

Now we are ready to construct the algorithm. We start by defining numbers $\alpha(n,k)$, $\beta(n,k)$ which satisfy assumptions (i), (iii) and (iv) in Proposition 3 (but not (ii)). First we prove the compatibility equations (10), (11):

Lemma 8. *Define*

$$\alpha(n,k) = f(k/n) = (2k/n) \wedge (1-2\epsilon). \tag{12}$$

Then for X satisfying (9), $\alpha(2n,k) \geq \mathbf{E}\alpha(n,X)$.

Proof. This follows from Jensen's inequality, since f is concave. □

The upper bound is more complicated. We would like $\beta(n,k)$ to be close to $\alpha(n,k)$, so that the algorithm is fast. Ideally, the difference should be exponentially small. This cannot be done over the whole interval $[0,1]$, since the Bernstein polynomials do not approximate f well near $1/2-\epsilon$, where it is not linear. To account for this, we also need a term of order $1/\sqrt{n}$, to be added if $k/n > 1/2 - 3\epsilon$. Finally, to control the speed of the algorithm for small p, we also want $\beta(n,k)$ and $\alpha(n,k)$ to be in fact equal if k/n is small.

To achieve this, consider the following auxiliary functions:

$$r_1(p) = C_1(p - (1/2 - 3\epsilon))_+, \quad r_2(p) = C_2(p - 1/9)_+.$$

The positive constants C_1 and C_2 will be determined later. Both functions are constant, equal to zero for p below a certain threshold, and increase linearly above the threshold. They are continuous and convex.

Lemma 9. *Define*

$$\beta(n,k) = f(k/n) + r_1(k/n)\sqrt{2/n} + r_2(k/n)\exp(-2\epsilon^2 n) \tag{13}$$

If $\epsilon < 1/8$ and X satisfies (9), then $\beta(2n,k) \leq \mathbf{E}\beta(n,X)\, \forall k, n$.

Proof. This amounts to proving

$$\begin{aligned} f(k/2n) - \mathbf{E}f(X/n) \quad &\leq \quad \mathbf{E}r_1(X/n)\sqrt{2/n} - r_1(k/2n)/\sqrt{2/(2n)} \\ &+ \quad \mathbf{E}r_2(X/n)\exp(-2\epsilon^2 n) - r_2(k/2n)\exp(-4\epsilon^2 n)\,. \end{aligned}$$

Since r_1 and r_2 are convex, $r_1(k/2n) \leq \mathbf{E}r_1(X/n)$ and $r_2(k/2n) \leq \mathbf{E}r_2(X/n)$, so it is enough to show

$$\begin{aligned} |f(k/2n) - \mathbf{E}f(X/n)| \quad &\leq \quad r_1(k/2n)(1 - 1/\sqrt{2})\sqrt{2/n} \\ &+ \quad r_2(k/2n)\exp(-2\epsilon^2 n)(1 - \exp(-2\epsilon^2 n))\,. \end{aligned}$$

If $k/2n \leq 1/8$, then $X/n \leq k/n \leq 1/4 \leq 1/2 - \epsilon$, so $f(X/n) = 2X/n$ for all values of X, so the left-hand side is in fact zero and the inequality holds.

If $1/8 \leq k/2n \leq 1/2 - 2\epsilon$, then we use the second part of Proposition 7 (the large deviation result). Thus, it suffices to show that

$$8 \leq r_2(k/2n)(1 - \exp(-2\epsilon^2 n)) .$$

But $r_2(k/2n) \geq C_2(1/8 - 1/9) = C_2/72$, so it is enough to choose

$$C_2 = 72(1 - \exp(-2\epsilon^2))^{-1} .$$

If $k/2n > 1/2 - 2\epsilon$, we use the first part of Proposition 7. It is enough then to show that $1 \leq r_1(k/2n)(1 - 1/\sqrt{2})$. But $r_1(k/2n) \geq C_1\epsilon$, so it is enough to choose $C_1 = \epsilon^{-1}(1 - 1/\sqrt{2})^{-1}$. This completes the proof of the lemma. □

We can now restate and prove

Theorem 1. For $\epsilon \in (0, 1/8)$, the function $f(p) = 2p \wedge (1 - 2\epsilon)$ has a simulation on [0,1], so that the number of inputs needed N satisfies $\mathbf{P}_p(N > n) \leq C\rho^n$, for all $n \geq 1$ and $p \in [0, 1/2 - 4\epsilon]$. The constants C and ρ depend on ϵ but not on p, and $\rho < 1$.

Proof. We use Proposition 3. First we prove that for $\alpha(n, k)$ and $\beta(n, k)$ defined in (12) and (13) and

$$g_n(x, y) = \sum_{k=0}^{n} \binom{n}{k} \alpha(n, k) x^k y^{n-k}, \quad h_n(x, y) = \sum_{k=0}^{n} \binom{n}{k} \beta(n, k) x^k y^{n-k}$$

conditions (i), (iii) and (iv) are satisfied for the subsequence $n_i = 2^i$. We have already proven (iv), and as discussed in the previous section, this implies that $g_n(p, 1 - p)$ is increasing and $h_n(p, 1 - p)$ is decreasing. By proposition 5, the Bernstein polynomials $g_n(p, 1-p)$ converge to f. Clearly $h_n(p, 1-p) - g_n(p, 1-p) \leq \sup_k(\beta(n, k) - \alpha(n, k)) \to 0$ as $n \to \infty$, so $h_n(p, 1 - p)$ also converges to f and we have proven (iii). (i) clearly holds for n large enough.

The remaining condition (ii) does not hold for $\alpha(n, k)$, $\beta(n, k)$, but as discussed in the previous section, we can get around this by defining

$$a(n, k) = \lfloor \alpha(n, k) \binom{n}{k} \rfloor / \binom{n}{k}, \quad b(n, k) = \lceil \beta(n, k) \binom{n}{k} \rceil / \binom{n}{k}. \tag{14}$$

Note that for $k/n < 1/9$, we have $\alpha(n, k) = \beta(n, k) = 2k/n$ so $\alpha(n, k)\binom{n}{k} = 2\binom{n-1}{k-1}$ is an integer, whence $a(n, k) = b(n, k)$.

The sequences $a(n,k)$, $b(n,k)$ satisfy conditions (i)-(iv), and the tail probabilities $\mathbf{P}_p(N > n) = h_n(p, 1-p) - g_n(p, 1-p)$ satisfy

$$\begin{aligned}
\mathbf{P}_p(N > n) &\leq \sum_{k=0}^{n} (\beta(n,k) - \alpha(n,k)) \binom{n}{k} p^k (1-p)^{n-k} + \sum_{k=n/9}^{n} 2p^k (1-p)^{n-k} \\
&\leq C_1 \sqrt{\frac{2}{n}} \sum_{k=\frac{n}{2}-3\epsilon n}^{n} \binom{n}{k} p^k (1-p)^{n-k} + C_2 e^{-2\epsilon^2 n} + \frac{2p^{n/9}}{(1-p)}. \quad (15)
\end{aligned}$$

The second term in (15) decays exponentially, and so does the third (we can use $4 \cdot 2^{-n/9}$ as an upper bound). For the first term, ignore the square root factor and look at the sum; it is equal to $\mathbf{P}(Y/n > 1/2 - 3\epsilon)$, where Y has binomial (n,p) distribution. Since $p \leq 1/2 - 4\epsilon$, a standard large deviation estimate (see [7]) guarantees that the first term in (15) is bounded above by $\exp(-2\epsilon^2 n)$, so it also decays exponentially in n.

Thus we do have $\mathbf{P}_p(N > n) \leq C\rho^n$ if n is a power of two. For general n, write $2^k \leq n < 2^{k+1}$. Then $\mathbf{P}_p(N > n) \leq \mathbf{P}_p(N > 2^k) \leq C\rho^{2^k} \leq C(\rho^{1/2})^n$. The proof is complete. □

Remark. Most of the proof works for a general linear function $f(p) = (ap) \wedge (1 - a\epsilon)$, for any $a > 0$. For integer a the whole proof works (with different constants). If a is not an integer then the only problem comes from rounding the coefficients; the rounding error introduced is bounded by $\sum_0^n p^k (1-p)^{n-k}$, which still decays exponentially, but the rate of decay approaches 1 as p approaches 0. In the next section we deduce a slightly weaker version of the result for general a as a consequence of the case $a = 2$.

Proposition 3 and Lemma 6 can also be used to obtain simulations for more general functions. The simulations are no longer guaranteed to be fast, but we do obtain *some* bounds for the tails of N:

Proposition 10. *Assume f satisfies $\epsilon < f < 1 - \epsilon$ on $(0,1)$. Then*

(i) *If f is Lipschitz, then it can be simulated with $\mathbf{P}_p(N > n) \leq D/\sqrt{n}$ for some uniform $D > 0$.*

(ii) *If f is twice differentiable, then it can be simulated with $\mathbf{P}_p(N > n) \leq D/n$ for some uniform $D > 0$.*

Remark. Neither of these conditions guarantees that N has finite expectation, though we do believe that this should be possible to achieve, at least for C^2 functions.

Proof. As in the proof of Theorem 1, it is enough to define numbers $\alpha(n,k)$, $\beta(n,k)$ which satisfy assumptions (i), (iii) and (iv) in Proposition 3; assumption

(ii) can then be achieved by rounding as described in Remark C of Proposition 3. We set

$$\begin{aligned} \alpha(n,k) &= f(k/n) - \delta_n \\ \beta(n,k) &= f(k/n) + \delta_n \end{aligned}$$

with $\delta_n \to 0$. Then (i) holds as soon as $\delta_n < \epsilon$ and (iii) holds because $g_n(p, 1-p) = Q_n(p) - \delta_n$, $h_n(p, 1-p) = Q_n(p) + \delta_n$, where Q_n are the Bernstein polynomials. It remains to check (iv), and as in the proof of Theorem 1, it is enough to do it for m, n powers of two, which amounts to checking that for hypergeometric X satisfying (9), we have $\alpha(2n, k) \geq \mathbf{E}\alpha(n, X)$ and $\beta(2n, k) \leq \mathbf{E}\beta(n, X)$. From Lemma 6,

$$\alpha(2n, k) - \mathbf{E}\alpha(n, X) \geq \delta_n - \delta_{2n} - C/\sqrt{2n}$$

if f is Lipschitz with constant C, and

$$\alpha(2n, k) - \mathbf{E}\alpha(n, X) \geq \delta_n - \delta_{2n} - C/(4n)$$

if f is twice differentiable and $|f''| \leq C$. The exact same inequalities hold for $\mathbf{E}\beta(n, X) - \beta(2n, k)$. Hence we can choose $\delta_n = (1 + \sqrt{2})C/\sqrt{n}$ in the Lipschitz case, and $\delta_n = C/(2n)$ in the twice differentiable case, and the proof is complete. □

4 Fast Simulation For Other Functions

We start with some facts about random variables with exponential tails.

Proposition 11. *Let $X \geq 0$ be a random variable. Then the following are equivalent:*

(i) *There exist constants $C > 0, \rho < 1$ such that $\mathbf{P}(X > x) \leq C\rho^x \, \forall x > 0$.*

(ii) $\mathbf{E}\exp(tX) < \infty$ *for some $t > 0$.*

If these hold, we say X has **exponential tails**.

Proof. Straightforward. □

Proposition 12. *Let $X_i \geq 0$ be i.i.d. with exponential tails, and let $N \geq 0$ be an integer-valued random variable with exponential tails. Then $Y = X_1 + \ldots + X_N$ has exponential tails.*

Proof. Take $t > 0$ such that $\mathbf{E}\exp(tX_1) < \infty$. Then we can find $k > 0$ such that $\rho = \mathbf{E}\exp(t(X_1 - k)) < 1$. Let $S_n = \sum_{i=1}^n X_i$. Then

$$\mathbf{P}(S_N > kn) \leq \mathbf{P}(N > n) + \mathbf{P}(S_n > kn).$$

The first term on the right decreases exponentially fast. To evaluate the second term, we use a standard large-deviation estimate:

$$\mathbf{P}(S_n > kn) \leq \exp(-tkn)\mathbf{E}\exp(tS_n) = (\mathbf{E}\exp(t(X_1 - k)))^n = \rho^n$$

so the second term also decreases exponentially fast and we are done. □

Remark. We do not assume that N is independent from the X_i's.

Proposition 13. *Constant functions* $f(p) = c \in [0, 1]$ *have a fast simulation on* $(0, 1)$.

Proof. For $f(p) = 1/2$, we can use Von Neumann's trick: toss coins in pairs, until we obtain 10 or 01; in the first case output 1, otherwise output 0 (if we obtain 11 or 00, we toss again). We need $2N$ tosses, where N has geometric distribution with parameter $p^2 + (1-p)^2$; this clearly has exponential tails (unless p is 0 or 1).

For any other constant c, write it in base two $c = \sum_{n=1}^{\infty} c_n 2^{-n}$ with $c_n \in \{0, 1\}$, generate fair coins using Von Neumann's trick, and toss them until we get a one. Output c_M, where M is the number of fair coin tosses. This scheme generates $f(p) = c$, and requires $X_1 + \ldots + X_M$ p-coin tosses, where X_i is the number of p-coin tosses needed to generate the i-th fair coin. All X_i have exponential tails and so does M, so Proposition 12 completes the proof. Note that the rate of decay of the tails depends on p but not on c; this will be used below. □

Proposition 14. *Let* $S, T \subset [0, 1]$.

(i) *If* f, g *have fast simulations on* S*, then the product* $f \cdot g$ *has a fast simulation on* S.

(ii) *If* f *has a fast simulation on* T *and* g *has a fast simulation on* S*, where* $g(S) \subset T$*, then* $f \circ g$ *has a fast simulation on* S.

(iii) *If* f, g *have fast simulations on* S *and* $f + g < 1 - \epsilon$ *on* S *for some* $\epsilon > 0$*, then* $f + g$ *has a fast simulation on* S.

(iv) *If* f, g *have fast simulations on* S *and* $f - g > \epsilon$ *on* S *for some* $\epsilon > 0$*, then* $f - g$ *has a fast simulation on* S.

Proof. (i) Let N_f, N_g be the number of inputs needed to simulate each function. We simulate f and g separately; if both algorithms output 1, we also output 1; otherwise, we output 0. This simulates $f \cdot g$ using $N_f + N_g$ inputs, which has exponential tails by Proposition 12.

(ii) We simulate g using its algorithm, then feed the results to the algorithm for f. We need $X_1 + \ldots + X_{N_f}$ inputs, where X_i are i.i.d. with the same distribution as N_g. This has exponential tails by Proposition 12.

(iii) We write $f + g = h \circ \psi$, where $h(p) = 2p$ and $\psi(p) = (f(p) + g(p))/2$. We proved in the previous section that h has a fast simulation on $[0, (1-\epsilon)/2]$. To simulate ψ, we simulate f and g separately to obtain binary variables B_f and B_g, then toss a fair coin; if the coin is heads, we output B_f, otherwise we output B_g. So ψ can be simulated using $N_f + N_g + N$ inputs, where N is the number of inputs needed to simulate a fair coin. Hence ψ also has a fast simulation, so (iii) follows from (ii).

(iv) Clearly f has a (fast) simulation iff $1 - f$ has one, so we can look at $1 - (f - g) = (1 - f) + g < 1 - \epsilon$. The conclusion then follows from (iii). □

Proposition 15. *If $a > 0$, $\epsilon > 0$, the function f has a fast simulation on S, and $af(p) < 1 - \epsilon$ on S, then $a \cdot f$ has a fast simulation on S.*

Proof. By Theorem 1, $2p$ has a fast simulation on $[0, 1/2-\epsilon)$. By the composition rule Proposition 14, (ii), $2^n p$ has a fast simulation on $[0, 1/2^n - \epsilon)$. For general $a > 0$, find n with $a < 2^n$ and write $ap = 2^n(a/2^n)p$. We know multiplication by 2^n has a fast simulation; so does multiplication by $a/2^n$, because constants smaller than 1 have a fast simulation. Hence their composition ap has a fast simulation on $[0, 1/a - \epsilon)$. We apply the composition rule Proposition 14, (ii) again to complete the proof. □

Proposition 16. *Let $f(p) = \sum_{n=0}^{\infty} a_n p^n$ with $a_n \geq 0$ for all n. Let $t \in (0, 1]$ such that $f(t) < 1$. Then f has a fast simulation on $[0, t - 2\epsilon], \forall\, \epsilon > 0$.*

Proof. Write

$$\frac{\epsilon}{t} f(p) = \sum_{n=0}^{\infty} (a_n t^n) \left(\frac{p}{t-\epsilon} \right)^n \left(\frac{t-\epsilon}{t} \right)^n \frac{\epsilon}{t}.$$

Since the terms $((t-\epsilon)/t)^n(\epsilon/t)$ are the probabilities of a geometric distribution, we can generate an $(\epsilon/t)f(p)$-coin as follows. First we obtain N with geometric distribution, so $\mathbf{P}_p(N = n) = ((t-\epsilon)/t)^n(\epsilon/t)$. Then we generate N i.i.d. $p/(t-\epsilon)$-coins (by Proposition 15, this can be done by a fast simulation), and we generate one $a_N t^N$-coin (since $f(t) < 1, a_N t^N < 1$). Finally, we multiply the $N + 1$ outputs as in Proposition 14, (i).

The number of coin tosses we need is $X + Y_1 + \ldots + Y_N + Z$, where X is the number of tosses required to obtain N, Y_i is the number of tosses required to generate the i-th $p/(t-\epsilon)$-coin, and Z is the number of tosses required to generate one (constant) $a_N t^N$-coin. Y_i have exponential tails by Proposition 15, and Z has exponential tails (whose rate of decay does not depend on the value of N) by Proposition 13.

The way we obtain N is we toss $(t-\epsilon)/t$-coins until we obtain a zero; hence X can itself be written as $X = W_1 + \ldots + W_N$, where W_i is the number of tosses required to generate a constant $(t-\epsilon)/t$-coin. Hence by Proposition 12, $(\epsilon/t)f(p)$ has a fast simulation.

Finally, $f = (t/\epsilon)(\epsilon/t)f$ has a fast simulation by Proposition 15. □

Proposition 17. *Let $f(p) = \sum_{n=0}^{\infty} a_n p^n$ have a series expansion with arbitrary coefficients $a_n \in \mathbf{R}$ and radius of convergence $R > 0$. Let $\epsilon > 0$ and $S \subset (0, 1)$ so that $\epsilon < f < 1 - \epsilon$ on S, and $\sup S < R$. Then f has a fast simulation on S.*

Proof. Separating the positive and negative coefficients, we can write $f = g - h$ where g, h are analytic with radius of convergence at least R, and have non-negative coefficients. They must also be bounded: $g \le M$ and $h \le M$, with $M = \sum_{n=0}^{\infty} |a_n| (\sup S)^n < \infty$. Then $g/2M$, $h/2M$ must have fast simulations on S by Proposition 16, so by Proposition 14, so does $2M(g/2M - h/2M)$. □

Proposition 18. *If f, g have fast simulations on S, are both bounded on S, $g > \epsilon$ on S, and $f/g < 1 - \epsilon$ on S for some $\epsilon > 0$, then f/g has a fast simulation on S.*

Proof. Let $M = \sup g$. Let $C \in (0, 1)$ and $h(p) = C/(1 - p) = \sum_0^\infty Cp^n$. By Proposition 16, this has a fast simulation on $(0, 1 - C - \epsilon/4M)$. We can replace $1 - p$ with p by switching heads and tails; hence $\psi(p) = C/p$ has a fast simulation on $(C + \epsilon/4M, 1)$. Set $C = \epsilon/4M$. Then ψ has a fast simulation on $(\epsilon/2M, 1)$ and so does $g/2M \in (\epsilon/2M, 1)$, so $\psi \circ g = \epsilon/(2g)$ has a fast simulation on S. So does the product $f \cdot (\psi \circ g) = (\epsilon/2)(f/g)$, and by Proposition 15 so does f/g, since we know it is bounded above by $1 - \epsilon$. □

Theorem 19. *Let f be a real analytic function on a closed interval $[a, b] \subset (0, 1)$, so f is analytic on a domain D containing $[a, b]$, and assume that $f(x) \in (0, 1)$ for all $x \in [a, b]$. Then f has a fast simulation on $[a, b]$.*

Proof. If D is the open disk of radius 1 centered at the origin, then f has a series expansion with radius of convergence 1 and the result follows from Proposition 17. For a general D, the idea of the proof is to map one of its subdomains to the unit disk, using a map which has a fast simulation.

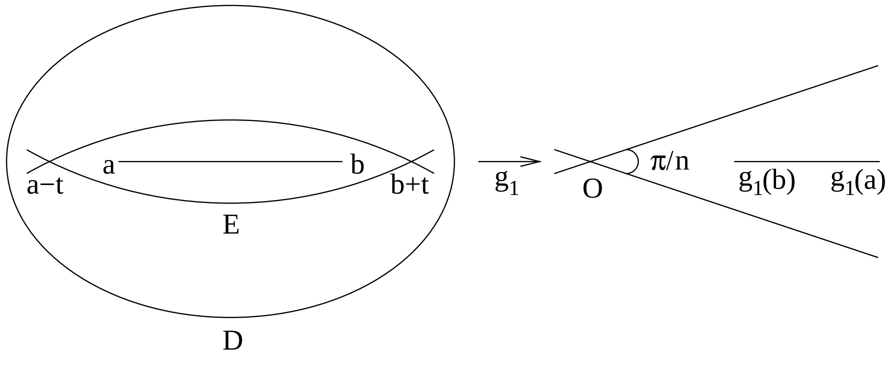

Figure 1: The map g_1.

Using a standard compactness argument, it is easy to show we can find a domain E so that $[a, b] \subset E \subset D$ and E is the intersection of two large open disks of equal radius. The centers of both disks are on the line $Re(z) = (a + b)/2$, located symmetrically above and below the real axis. The boundaries of the disks intersect on the real axis at the points $a - t$ and $b + t$ for some small $t > 0$.

If we make the radius of the disks large enough, we may assume that the angle between the disks is π/n for some large integer n.

We shall use a Möbius map of the form $(pz+q)/(rz+s)$ to map those disks into half-planes. Fix $c > 0$. The map

$$g_1(z) = \frac{c}{z-(a-t)} - \frac{c}{(b+t)-(a-t)} \tag{16}$$

maps the boundaries of the disks into lines going through the origin, so it maps E to the domain between those two lines contained in the positive half-plane $Re(z) > 0$. The angle between the two lines is π/n, so the map g_1^n maps E to the positive half-plane.

The map $g_2(z) = 1 - 2/(1+z)$ maps the positive half-plane to the unit disk, so $g_2 \circ g_1^n$ maps E to the unit disk. Hence $f \circ (g_1^n)^{-1} \circ (g_2)^{-1}$ is real analytic on the unit disk (it is easy to check that the inverses of g_1^n and g_2 are analytic on their respective domains), so it has a fast simulation on any closed interval contained in $(0,1)$. It remains to check that $g_2 \circ g_1^n$ maps $[a,b]$ to such an interval, and that it has a fast simulation. Then it follows from Proposition 14, (i) that f also has a fast simulation.

For sufficiently large c, the function g_1 maps the interval $[a,b]$ to the interval $[g_1(b), g_1(a)]$ where $1 < g_1(b)$. Hence $1/g_1$ maps $[a,b]$ to some closed subinterval of $(0,1)$, and by Proposition 18 it has a fast simulation (as the ratio of two linear functions). Clearly, so does $1/g_1^n$. Finally, we can write $g_2 \circ g_1^n = g_3 \circ (1/g_1^n)$, where $g_3(z) = g_2(1/z) = 1-(2z)/(1+z)$ also has a fast simulation, by the same Proposition 18. This completes the proof. □

5 Necessary Conditions For Fast Simulations

Proposition 20. *Assume f has a fast simulation on an open set $S \subset (0,1)$. Then f is real analytic on S.*

Proof. Consider a fast algorithm, fix p and let $f_n(p)$ be the probability that it outputs 1 after exactly n steps. Then $f = \sum_1^\infty f_n$ and

$$0 \le f(p) - \sum_1^n f_i(p) = \sum_{n+1}^\infty f_i(p) \le C\rho^n \quad \forall n \ge 0$$

for some constants $C > 0, \rho < 1$. Pick any B with $1 < B < 1/\rho$. Since f_n are polynomials, $f_n(z)$ is well-defined for any complex z. We shall prove below that we can find $\epsilon > 0$ so that for any complex z and positive integer n,

$$|f_n(z)| \le B^n f_n(p) \quad \text{if } |z-p| < \epsilon. \tag{17}$$

Then for any $m > n$ and $z \in B(p,\epsilon)$ (the open ball with center p and radius ϵ), we have

$$\begin{aligned}|\sum_{i=n+1}^{m} f_i(z)| &\leq \sum_{i=n+1}^{m} |f_i(z)| \\ &\leq \sum_{i=n+1}^{m} B^i f_i(p) \leq \sum_{i=n+1}^{\infty} B^i C\rho^{i-1} = (B\rho)^n BC/(1-B\rho).\end{aligned}$$

Hence the sequence $\{\sum_1^n f_i\}$ is Cauchy on $B(p,\epsilon)$, so it converges uniformly on $B(p,\epsilon)$ to a limit which is analytic by a standard theorem (see [1], p.176, Theorem 1). Hence f is real analytic.

To prove (17), note that f_n can be written as $f_n(z) = \sum_{k=0}^{n} a_{n,k} z^k (1-z)^{n-k}$ with $a_{n,k} \geq 0$. Since $|z-p| < \epsilon$ we have $|z| < p+\epsilon$ and $|1-z| < 1-p+\epsilon$. Choose ϵ so $p+\epsilon < Bp$ and $1-p+\epsilon < B(1-p)$. Then

$$|z^k(1-z)^{n-k}| \leq (p+\epsilon)^k (1-p+\epsilon)^{n-k} \leq B^n p^k (1-p)^{n-k}$$

and

$$|\sum_{k=0}^{n} a_{n,k} z^k (1-z)^{n-k}| \leq \sum_{k=0}^{n} a_{n,k} |z^k(1-z)^{n-k}| \leq B^n \sum_{k=0}^{n} a_{n,k} p^k (1-p)^{n-k}$$

as desired. □

Proposition 21. *Assume $S \subset [0,1]$ is closed and f has a fast simulation on S. Then the number of inputs N has uniformly bounded tails: there exist constants C, ρ which do not depend on p, so $\mathbf{P}_p(N > n) \leq C\rho^n$, $\forall p \in S$.*

Proof. Let $g_n(p) = \mathbf{P}_p(N > n)$. Just as in Proposition 20, g_n can be written as $g_n(z) = \sum_{k=0}^{n} a_{n,k} z^k (1-z)^{n-k}$ with $a_{n,k} \geq 0$, so for any $p \in (0,1)$ and $B > 1$ we can find $\epsilon > 0$ so

$$|g_n(z)| \leq B^n g_n(p) \quad \text{if } |z-p| < \epsilon. \tag{18}$$

For any $p \in S \bigcap (0,1)$ we have $g_n(p) \leq C_p \rho_p^n$ for some $C_p > 0, \rho_p < 1$. Setting $B = \rho_p^{-1/2}$ in (18) we obtain that there exists $\epsilon_p > 0$ so

$$g_n(z) \leq C_p \rho_p^{n/2} \quad \text{if } z \in (p-\epsilon_p, p+\epsilon_p).$$

The intervals $(p-\epsilon_p, p+\epsilon_p)$ cover S. Since S is closed it is compact, so we can find a finite subcover $(p_i - \epsilon_{p_i}, p_i + \epsilon_{p_i}), 1 \leq i \leq N$. Then we can set

$$C = \max C_{p_i}, \quad \rho = \max \rho_{p_i}^{1/2}.$$ □

Remark. This also shows that if a function has a simulation on some $S \subset (0,1)$, then the set of p where the simulation is fast is open in S.

Proposition 22. *Assume f has a simulation on an open set $S \subset (0,1)$, such that the number of inputs needed N has finite k-th moment on S, and furthermore the tails of the moments decrease uniformly:* $\lim_{n\to\infty} \mathbf{E}_p N^k 1(N > n) = 0$ *uniformly in $p \in S$. Then $f \in C^k(S)$ (i.e., f has k continuous derivatives on S).*

Proof. Let f_n be defined as in Proposition 20. Since $f = \sum_1^\infty f_n$, it is enough to prove that the series $\sum_1^\infty f_n^{(k)}$ converges uniformly on S. We shall prove that $|f_n^{(k)}| \le C n^k f_n$ for a uniform constant C. Then

$$\sum_{n=m}^{\infty} |f_n^{(k)}| \le \sum_{n=m}^{\infty} C n^k f_n = C\mathbf{E}_p N^k 1(N > m-1)$$

converges to zero uniformly as $m \to \infty$, so the series is Cauchy and we are done. To prove the required inequality, recall that $f_n(p) = \sum_{i=0}^n a_{n,i} p^i (1-p)^{n-i}$ with $a_{n,i} \ge 0$. Write $[i]_j = i(i-1)\dots(i-j+1)$. From Leibniz' formula for the derivative of a product,

$$\begin{aligned}
|(p^i(1-p)^{n-i})^{(k)}| &= |\sum_{j=0}^{k} \binom{k}{j} (p^i)^{(j)}((1-p)^{n-i})^{(k-j)}| \\
&= |\sum_{j=0}^{k} \binom{k}{j} [i]_j p^{i-j} [n-i]_{k-j} (1-p)^{n-i-(k-j)} (-1)^{k-j}| \\
&\le \sum_{j=0}^{k} (k!) n^k p^i (1-p)^{n-i} / \min(p, 1-p)^k \\
&\le C n^k p^i (1-p)^{n-i}
\end{aligned}$$

for $C = k(k!)/\inf_{q\in B} \min(q, 1-q)^k$, where the inf is taken over some small neighborhood B of p. It follows that $|f_n^{(k)}| \le C n^k f_n$ on S. □

Proposition 23. *Assume f has a simulation on a closed interval $I \subset (0,1)$, such that the number of inputs needed N has $\sup_{p\in I} \mathbf{E}_p(N) < \infty$. Then f is Lipschitz over I.*

Proof. We are given that $\mathbf{E}_p N = \sum_1^\infty n f_n \le C < \infty$. Since I is closed, $I \subset (\epsilon, 1-\epsilon)$ for some ϵ. As in the previous proposition, we obtain $|f_n'| \le n f_n / \min(\epsilon, 1-\epsilon)$. Hence $|\sum_1^n f_i'| \le C/\min(\epsilon, 1-\epsilon)$ so

$$|\sum_1^n f_i(p) - \sum_1^n f_i(q)| \le |p-q| C/\min(\epsilon, 1-\epsilon).$$

Letting $n \to \infty$ finishes the proof. □

6 An Approximate Algorithm For Doubling

The methods described in the previous sections are essentially constructive. Proposition 3 gives a recipe for constructing an algorithm, given an approximation; all that is needed is an ordering of all binary words of length n with k 1's.

In the particular case of the function $f(p) = 2p$, there exists an extremely simple algorithm. It also works for any $p \in (0, 1/2)$; there is no need to bound the function away from 1. The catch is that it is approximate: it outputs 1 with probability very close to $2p$, with the error decaying exponentially in the number of steps. This must be, of course; the Keane - O'Brien results show that we couldn't have an **exact** algorithm with these properties. However, in practice, an approximate result may suffice.

Proposition 24. *Let $p < 1/2$ and consider an asymmetric simple random walk $S_n = X_1 + \ldots + X_n$, with $\mathbf{P}_p(X_i = 1) = p = 1 - \mathbf{P}_p(X_i = -1)$. Let A_n be the event that $\max(S_1, \ldots, S_n) \geq 0$. Then $\mathbf{P}_p(A_n) = \sum_{k=0}^{n}(2k/n \wedge 1)\binom{n}{k}p^k(1-p)^{n-k} = Q_n(p)$, where Q_n is the n-th Bernstein polynomial of the function $f(p) = 2p \wedge 1$.*

Proof. We need to show that the number of paths with k positive steps among the first n steps, and $\max(S_1, \ldots, S_n) \geq 0$, is $(2k/n \wedge 1)\binom{n}{k}$. For $k > n/2$, this is obvious. For $k \leq n/2$, $(2k/n)\binom{n}{k} = 2\binom{n-1}{k-1}$ and the result follows from the reflection principle (see, for example, [3], p. 197). ☐

Since f is piecewise linear, its Bernstein polynomials converge to it exponentially fast (except at $p = 1/2$), so we obtain the following

Algorithm. Run an asymmetric simple random walk $S_n = X_1 + \ldots + X_n$, with $\mathbf{P}_p(X_i = 1) = p = 1 - \mathbf{P}_p(X_i = -1)$ for at most n steps. If the walk ever reaches non-negative territory ($S_k \geq 0$ for some $1 \leq k \leq n$), output 1. Otherwise, stop after n steps, output 0.

A standard large deviation estimate (see [7]) shows that if $p < 1/2$, the probability of outputting 1 is $2p - \epsilon$, where $0 \leq \epsilon \leq 2\exp(-2n(1/2 - p)^2)$.

See [5] for another construction of an approximate doubling algorithm.

7 Continuous Functions Revisited

In this section we use Proposition 3 to simulate any continuous function f that satisfies $\epsilon < f \leq 1 - \epsilon$ on $(0, 1)$ for some $\epsilon > 0$. Our proof is simpler than the original proof of Keane and O'Brien in [8]. However, their argument is more general since it does not assume that f is bounded away from 0 and 1. We will use the following theorem of Pólya:

Theorem 25. *Let $q(x,y)$ be a homogenous polynomial with real coefficients satisfying $q(x,y) > 0\,\forall x > 0, y > 0$. Then for some nonnegative integer n, all coefficients of $(x+y)^n q(x,y)$ are non-negative.*

See [6], p. 57-59 for a proof. This clarifies the connection between the partial order $\preceq$ in Definition 2 and the pointwise partial order. It says that if $q(x,y) < r(x,y)$ for all $x, y > 0$, then $(x+y)^n q(x,y) \prec (x+y)^n r(x,y)$ for some n.

Theorem 26. (Keane-O'Brien [8]) *Let $\epsilon > 0$ and suppose that $f : (0,1) \mapsto [\epsilon, 1-\epsilon]$ is continuous. Then f admits a terminating simulation.*

Proof. Let i satisfy $2^{-i} < \epsilon/4$. By Proposition 5, we can approximate $f - 3 \cdot 2^{-i}$ by a Bernstein polynomial q_{m_i} of sufficiently high degree m_i with error smaller than 2^{-i}. More precisely,

$$q_{m_i}(x,y) = \sum_{k=0}^{m_i} \binom{m_i}{k} \Big(f(k/m_i) - 3 \cdot 2^{-i} \Big) x^k y^{m_i - k}$$

will satisfy $f(p) - 4 \cdot 2^{-i} < q_{m_i}(p, 1-p) < f(p) - 2 \cdot 2^{-i}$ for all $p \in (0,1)$.

The sequence $q_{m_i}(p, 1-p)$ is increasing in i, so

$$q_{m_i}(x,y)(x+y)^{m_{i+1} - m_i} < q_{m_{i+1}}(x,y) \quad \forall x, y > 0\,.$$

By Theorem 25,

$$q_{m_i}(x,y)(x+y)^{m_{i+1} - m_i + s_i} \prec q_{m_{i+1}}(x,y)(x+y)^{s_i}$$

for some integer $s_i \geq 0$. Thus if we define $n_1 = m_1$ and more generally, $n_i = m_i + (s_1 + \ldots + s_{i-1})$, then the homogenous polynomials

$$g_{n_i}(x,y) = q_{m_i}(x,y)(x+y)^{n_i - m_i}$$

satisfy conditions (i), (iii) and (iv) in Proposition 3 along the subsequence $\{n_i\}$. Condition (ii) is easily obtained by the rounding process described in Remark C after Proposition 3. By Remark B there, once we have g_n for the subsequence $n = n_i$, we can define it for all n. A similar construction can be used to define approximations from above h_n. (In fact these approximations will require another sequence $\{s'_i\}$ analogous to $\{s_i\}$ above, and for consistency we need to use $\max\{s_i, s'_i\}$ in both approximations.) Hence by Proposition 3, f has a terminating simulation algorithm. □

8 Open Problems

Theorem 2 does not settle the issue of what happens near 0 and 1, or on the boundary of the domain of analyticity of a function. An interesting example is the square root function $f(p) = \sqrt{p}$. Our methods provide fast simulations on any interval $(\epsilon, 1]$, but if p is allowed to take any value in $(0,1)$, the best result we are aware of is the one in [10], where the authors construct a simulation using a random walk on a ladder graph. Estimates for the tails of the number of inputs needed N are then given by return probabilities for a simple random walk, so $\mathbf{P}_p(N > n)$ decays like $n^{-1/2}$. We do not know whether one can do better.

Question 1. Is there an algorithm that simulates $\sqrt{p}$ on $(0, 1)$, for which the number of inputs needed has finite expectation for all p?

Remark. Entropy considerations (see [2], page 43) imply that if an algorithm as in Question 1 exists, then the expectation of the number of inputs cannot be uniformly bounded on $(0, 1)$. Indeed, this expectation must be at least $H(\sqrt{p})/H(p)$, where $H(p) = -p\log(p) - (1-p)\log(1-p)$ is the entropy function.

Question 2. Let $J \subset (0, 1)$ be a closed interval and let $f : J \mapsto (0, 1)$ be continuous. Suppose that we have a simulation algorithm that takes as input a sequence $\{X_i\}$ of i.i.d. p-coins and produces a sequence of i.i.d. $f(p)$-coins. The *rate* of the algorithm (when it exists) is defined to be the limit as $n \to \infty$ of $1/n$ times the expected number of $f(p)$ coins produced from the first n inputs. The rate can never exceed the entropy ratio $H(p)/H(f(p))$, see [2]. Given J and f, are there simulation algorithms with rates arbitrarily close to the entropy ratio, uniformly for all $p \in J$?
A positive answer is known for constant f: for $f(p) \equiv 1/2$ variants of the von Neumann scheme (see [4, 12]) will do, and other constants follow from combining these with [9]. However, for nonconstant f (except the identity and $f(p) = 1 - p$) the situation is unclear; a good example to ponder is $f(p) = p^2$.

We would also like to know whether Proposition 22 can be improved.
Question 3. Is it true (possibly subject to some technical conditions) that a function has a simulation where the number of inputs has uniformly bounded k-th moment, if and only if it has k continuous derivatives?

Acknowledgement. We are grateful to Jim Propp for suggesting the simulation problem to us, and to Omer Angel and Elchanan Mossel for helpful discussions.

References

[1] Ahlfors, L. V. (1978) *Complex Analysis*, Third Edition. McGraw-Hill, New York.

[2] Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory.* Wiley, New York.

[3] Durrett, R. (1996) *Probability: Theory and Examples.* Duxbury Press.

[4] Elias, P. (1972) The efficient construction of unbiased random sequence, *Ann. Math. Stat.* **43**, 865–870.

[5] Glynn, P. W. and Henderson, S. G. (2003) Nonexistence of a Class of Variate Generation Schemes. *Oper. Res. Let.* **31**, 83–89.

[6] Hardy, G. H., Littlewood, J. E., and Pólya, G. (1959) *Inequalities.* Cambridge University Press.

[7] Hoeffding, W. (1963) Probability Inequalities For Sums Of Bounded Random Variables. *J. Amer. Stat. Assoc.*, **58**, 13–30.

[8] Keane, M. S. and O'Brien, George L. (1994) A Bernoulli Factory. *ACM Transactions on Modeling and Computer Simulation*, **4**, 213–219.

[9] Knuth, D. E. and Yao, A. C. (1976) The complexity of nonuniform random number generation. *Algorithms and complexity* (Proc. Sympos., Carnegie-Mellon Univ., Pittsburgh, Pa., 1976), 357–428. Academic Press, New York.

[10] Mossel, E. and Peres, Y. (2002) New coins from old: computing with unknown bias. *Preprint*, available at `http://front.math.ucdavis.edu/math.PR/0304143`

[11] von Neumann, J. (1951) Various techniques used in connection with random digits. *National Bureau of Standards Applied Math Series* **12**, 36–38.

[12] Peres, Y. (1992). Iterating von Neumann's Procedure for Extracting Random Bits, *Ann. Stat.*, **20**, 590–597.

[13] Tikhomirov, V. M. (1990) *Approximation Theory*, in Analysis II, Encyclopedia of Mathematical Sciences, Vol. 14 (Ed. Gamkrelidze, R. V.), Springer-Verlag.