Efficient sampling from perpetuities using coupling from the past

Omar Fawzi

September 3, 2007

1 Introduction

In numerous fields, one needs to get samples of a set of objects according to a given distribution. Sometimes, the structure of this set is complicated so one cannot enumerate the objects efficiently. In these cases, an ergodic Markov chain is devised with the wanted distribution as a stationary distribution, the state space of these chains is usually huge but it is easy to compute transitions. Monte Carlo Markov Chain simulation consists on beginning in an arbitrary state and running the transition a sufficiently long time so that we get close enough to the limit distribution, in this way we get an **approximate** sampling technique.

Propp and Wilson in [12] devise a way to get a **perfect** sample from the stationary distribution of a generic finite ergodic Markov chain in expected **finite** time, this method is known as coupling from the past (CFTP).

In this report, we will be using this method to get exact samples from particular distributions given as a solution of a stochastic equation. We will begin with the distribution of a random variable X for which

$$X \stackrel{\mathcal{L}}{=} (1+X)U \tag{1}$$

where U is a uniform independent of X. This distribution is known as the Dickman distribution and it appears for example in analytic number theory and in the analysis of the Quickselect algorithm ([6] and [7]). For example, in [7], it is shown that the Dickman function is the limit distribution of $\frac{C_{n,m}-n}{n}$ provided $m = o(n)$ where $C_{n,m}$ is the (random) number of comparisons done by Hoare's selection algorithm to find the m-th smallest element in a list of n elements.

We will try to develop a general way for getting samples from distributions defined by a stochastic equation using coupling from the past. Technically speaking, we will begin with equations $X \stackrel{\mathcal{L}}{=} U^{\alpha}(1+X)$ which are called Vervaat perpetuities, and then more general perpetuities $X \stackrel{\mathcal{L}}{=} AX + B$, with some constraints on random variables A and B. Devroye in [3] developed methods for sampling from Vervaat perpetuities using rejection sampling, but these methods are quite complicated so our objective is to develop easier techniques that are more efficient and easier to generalize to different fixed-point equations. Also Kendall and Thönnes in [9] and Fill and Huber in [4] have developed methods using coupling from the past but we haven't seen a published paper discussing the problem in detail.

The two main problems we have to tackle to achieve our goal is the fact that the state spaces of the chains we are considering are unbounded and the fact that we have to find appropriate coalescing couplings.

2 Preliminaries

2.1 Notations

The set of integers will be denoted by $\mathbb{Z}$, and the set of reals $\mathbb{R}$. For the set of nonnegative integers we write $\mathbb{Z}_+$ and $\mathbb{R}_+$ for the set of nonnegative reals. The indicator function of a set A, that is the function whose value is 1 on A and 0 everywhere else will be denote $\mathbb{I}_A$.

We will be writing $\mathbb{P}\{A\}$ for the probability of an event A, and $\mathbb{E}\{X\}$ for the expected value of a random variable X. The law of a random variable X will be denoted by $\mathcal{L}(X)$. Also, when we have a probability measure P having a distribution function F, and a random variable X having distribution P, we will write $X \sim P$ or $X \sim F$. If two random variables X and Y follow the same law, we write $X \stackrel{\mathcal{L}}{=} Y$.

Here are some of the standard probability distributions we will be using:

- The Geometric distribution with parameter $0 < p < 1$, is a distribution on $\mathbb{Z}_+$ defined by $\mathbb{P}\{X = n\} = (1-p)p^n$ for $n \geq 0$.
- The Poisson distribution is the distribution on $\mathbb{Z}_+$ defined by $\mathbb{P}\{X = n\} = \frac{e^{-1}}{n!}$ for $n \geq 0$.
- The Uniform distribution is the distribution on $[0, 1]$ having distribution function $x \mapsto x\mathbb{I}_{[0,1]}$, and density $x \mapsto 1\mathbb{I}_{[0,1]}$
- The Exponential distribution is the distribution on $\mathbb{R}_+$ with distribution function $x \mapsto \mathbb{I}_{\mathbb{R}_+}(1 - e^{-x})$, and density function $x \mapsto \mathbb{I}_{\mathbb{R}_+}e^{-x}$.

And about the convergence of random variables, we say that X_n converges to X in distribution when the sequence of distribution functions F_n of X_n converges to the distribution function F of X for every point. We say that X_n converges to X almost surely when $X_n(\omega)$ converges to $X(\omega)$ for every $\omega \in A$ where A is an event with probability 1.

2.2 Sampling

In our context, we suppose we have access to a random number generator that produces independent uniform random variables, and we also suppose we do the calculations with infinite precision. Provided we have this kind of random number generator, the samples that our procedure will output will have the correct distribution.

Of course this is not true in practice. The random number generator are not actually ideally random, and we cannot calculate with infinite precision, but our focus is to build random variables with complicated distributions from basic building blocks, we have to suppose the random number generators are perfect if our objective is exact sampling. Constructing efficient pseudo-random number generators is another subject.

2.3 Markov chains

We will be using Markov chains, so let us have a quick review. A Markov chain is a sequence X_n of random variables taking values in a set of states S, and such that X_{n+1} depends only on X_n and not on $X_0, \dots X_{n-1}$. This means that

$$\mathbb{P}\{X_{n+1} = x | X_1 = x_1, \dots X_n = x_n\} = \mathbb{P}\{X_{n+1} = x | X_n = x_n\}.$$

A Markov chain is defined by giving an initial distribution on S (the distribution of X_0) and a transition matrix P, whose elements p_{ij} represent the probability of going from state i to state j. We will be writing $p_{ij}^{(n)}$ the probability of going from state i to state j in n steps.

We will also be representing the transition function as a random mapping, for $x \in S, \phi(x)$ is the random next state. And when we need to highlight the randomness used in the mapping, we will represent the next state by a deterministic function $x \mapsto \phi(x, U)$ taking as an argument a random variable (that will often be a uniform random variable). Here are some basic definitions about Markov chains

Definition 1.
- An irreducible chain is a chain such that for all pairs of states i and j, starting a walk from i, there is a positive probability of reaching j i.e., there exists n such that $\mathbb{P}\{X_n = j | X_0 = i\} > 0$.
- An aperiodic chain is a chain such that for all states i, j $\gcd\{n : \mathbb{P}\{X_n = j | X_0 = i\} > 0\} = 1$.
- An ergodic chain is one that is both irreducible and aperiodic.
- A stationary distribution π for a Markov chain is a distribution that is invariant after a move of the chain. If $X_0 \sim \pi$ then $X_1 \sim \pi$.
- The mean recurrence time of state i is $\mathbb{E}\{T_i\}$ where T_i is the random variable $T_i = \min\{n : X_n = i | X_0 = i\}$.

Ergodic chains are important because they have the following property:

Property 1. *An ergodic chain has a unique stationary distribution π, and for any initial distribution, X_n converges in distribution to π. This means that for every state i, $\mathbb{P}\{X_n = i\} \xrightarrow[n\to\infty]{} \pi(i)$. Moreover, the mean recurrence time $\mathbb{E}\{T_i\} = \frac{1}{\pi(i)}$.*

Markov chains are very useful in sampling, in fact, when we need to get a random combinatorial object, it is often easy to construct a Markov chain that makes moves between such objects. For example, consider we want to generate a random matching of a graph. We see that it is not easy to have an enumeration of the matchings from which we could sample, as if they were only numbers. But it turns out it is easy to build a Markov chain whose state space is the space of matchings and whose stationary distribution is uniform on all the matchings of this graph. The transition from a matching M is basically to choose an edge e of the graph randomly, to remove it if it's already in M, to add it to our matching M if we can, or else to stay with M. Now that we have our Markov chain, we can get approximate samples by running the Markov chain long enough, and one can calculate this time depending on the desired approximation by studying the mixing time of this chain.

This technique is called Monte Carlo Markov Chain and it has a lot of applications, not only in computer science but also in statistical physics. The technique was originally conceived to approximate integrals (viewed as expectations), by using the law of large numbers. An important example is the Gibbs sampler which is heavily used in probabilistic inference. Sampling is also related to approximate counting, if we can sample the matchings in a graph, then we can approximate the number of matchings in this graph. Sampling is also very important in statistical physics simulations.

2.4 Rejection sampling

Another sampling method we will be using is based on the density of our distribution. Suppose we have a distribution with density function f. Then it is known that the distribution of X has density f if (X, Y) is uniform on the surface $S_f = \{(x, y) \in \mathbb{R}^2 | 0 \leq y \leq f(x)\}$. Reciprocally, if X has density f, and $Y = Uf(X)$ where U is an independent uniform random variable, then (X, Y) is uniform on the same surface. Suppose now we have a complicated density function f that we want to sample from. It is not easy to get a uniform random variable on S_f. Now consider another function g that verifies $f(x) \leq g(x)$ and is equal to $g(x) = c \times h(x)$ where c is a constant and h is an "easy" density function, i.e. a density function from which we can sample (for example an exponential or a uniform). Note that it is easy to get a sample uniform on the surface S_g defined by $g = c \times h$, in fact just get X from the density h and $Y = Uc \times h(X)$, U being a uniform on $[0, 1]$.

We get two possible cases, either $(X, Y) \in S_f$ and we output (X, Y) or $(X, Y) \notin S_f$ and we reject the point (X, Y). Using this procedure, called rejection sampling, we get a uniform point on S_f, and by taking the x-coordinate we get a sample with distribution having density f.

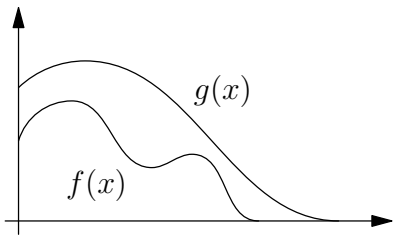

Figure 1: *Rejection sampling*

2.5 Coupling from the past

Coupling from the past (CFTP) is a method for getting an exact sample from the stationary distribution of a Markov chain. Consider starting chains at time $-T$ from all the states of the chain, and whenever two of them end up in the same state, they stay together. Suppose that when we reach time 0, all the chains have reached the same state s. This means that if we use the same randomness, and we start from any state from time $-t$ for any $-t \leq -T$, then we will end up in that same state s, simply because at time $-T$ we have to encounter one of the states of the Markov chain, and then we will follow the same route. The idea of CFTP is to start from an arbitrary time in the past $-t$ and run all the chains until time 0, if we have coalescence, then the single state reached at time 0 by all the chains is our sample. Else, we go back say by one starting at $-t-1$ and again check for coalescence at time 0. Note that it is very important to do the same transitions as the ones we did in the previous unsuccessful step, that is keep the randomness used at each time.

Now let us make things more formal. Suppose we have a finite set of states S, and random maps f_t describing the transitions at time t. Instead of going forward as in MCMC for example, we will at each step go backwards one step more, and always look at the state at time 0. We denote by F^0_{-t} the mapping corresponding to the transition from time $-t$ to time 0, that is

$$F^0_{-t} = f_{-1} \circ \cdots \circ f_{-t}.$$

For example $F^0_{-1}(x)$ is a random state we get at step 0 when we start with state x at time -1. Note that calculating $F^0_{-t}(x)$ for a certain x is harder than simply going forwards, in fact we cannot compute $F^0_{-t}(x)$ from $F^0_{-t+1}(x)$. Instead we have to keep $F^0_{-t+1}(y)$, for **all states** y, so that we calculate $f_t(x)$, and then apply F^0_{-t+1} to $f_t(x)$. But we'll get the reward later. To get a sample from the stationary distribution of our Markov chain, we can calculate for any state x, $F^0_{-\infty}(x)$ which will be well distributed. The idea of couping from the past is that we do not need to go to time $-\infty$ to get this value. In fact suppose that for some T, the mapping F^0_{-T} is a constant mapping. This means that if we start from time $-T$ in any state, we will end up in a single same state s. But then consider $F^0_{-T-1}(x) = F^0_{-T}(f_{-T-1}(x)) = s$, and thus we get $F^0_{-\infty}(x) = s$.

So it is now clear how we will proceed, compute the mapping F^0_{-t} for increasing values of t until it is constant and then output this constant. The question now is how we can be sure our mapping will ever be constant (we will be calling this event coalescence). This depends on the coupling used. Note that we did not mention anything about the relations between $f_t(x)$ and $f_t(y)$, we know that both of them have a distribution described by the Markov chain but how are they related. Are they independent or dependent ? Actually this does not matter, we can couple these two random variables as we want. Let us show that in the case of independent choices we have coalescence with probability one.

In fact because the chain is ergodic, for all states x, y there exists an $M_{x,y}$ such that $n \geq M_{x,y}, p^{(n)}_{xy} > 0$. Let $M = \max M_{x,y}$, so the probability that F^0_{-M} is constant is greater than $\prod_{s \in S} p^{(M)}_{sx}$ and hence positive. The same holds for F^{-M}_{-2M}, and $F^{-kM}_{-(k+1)M}$. Now because each of these mappings are independent, then the probability that there does not exist a T such that F^0_{-T} is constant is 0.

Let us illustrate CFTP with a simple example. The standard example given to illustrate this method is a random walk on n points. Consider we have n states numbered from 1 to n. When we are at state k we move to $\max(k-1, 1)$ with probability $\frac{1}{2}$ or $\min(k+1, n)$ with probability $\frac{1}{2}$. To define our update, the most natural idea, is to flip a coin that will be the same for all states i.e., either everybody goes up, or everybody goes down.

$$\phi(k, U) = \begin{cases} \max(k-1, 1) & \text{if } U \leq \frac{1}{2} \\ \min(k+1, n) & \text{if } U > \frac{1}{2} \end{cases}$$

It is clear that the uniform distribution on the states is the stationary distribution of this Markov chain. Fig. 2 shows a sample run of the procedure on this Markov chain.

Note that the algorithm as stated needs a lot of memory. In fact we need to store functions over the set of states, and often the state space is very large, and that is why we use MCMC methods to get samples. A possible solution is to ensure that we have a **monotone** coupling. This means that if $x \leq y$ then $f_t(x) \leq f_t(y)$ with probability 1. In our example, the coupling was monotone, graphically this means two arrows never cross each other. In this case it is possible to keep track only of the maximum state s_{max} and the minimum state s_{min}, because we know that if $F^0_{-t}(s_{min}) = F^0_{-t}(s_{max})$, then F^0_{-t} is a constant function.

Monotonicity is all the more important for us because our state space will not only be big, it will be infinite, in fact our state space is $\mathbb{R}_+$.

For more details on CFTP, see the original paper [12] and Wilson's web site on perfect sampling [15].

2.6 Stochastic equations

We are interested in a type of distributions that are described by an equation. For example, we know that if X has distribution μ, and if A and B are independent of X with some fixed distribution then $AX + B$ has distribution μ. We assume that A and B are positive random variables and in all the case we will consider X will be positive. Which conditions on A and B ensure the existence and uniqueness of the solution to equation $X \stackrel{\mathcal{L}}{=} AX + B$? This kind of problem is solved using fixed point arguments. As in [13] we introduce a metric (Mallow metric) on distribution functions that have finite second moment $\mathbb{E}\{X^2\} < \infty$:

$$d_2(F, G) = \inf_{X \sim F, Y \sim G} ||X - Y||_2 = \inf_{X \sim F, Y \sim G} \mathbb{E}((X - Y)^2)^{1/2}$$

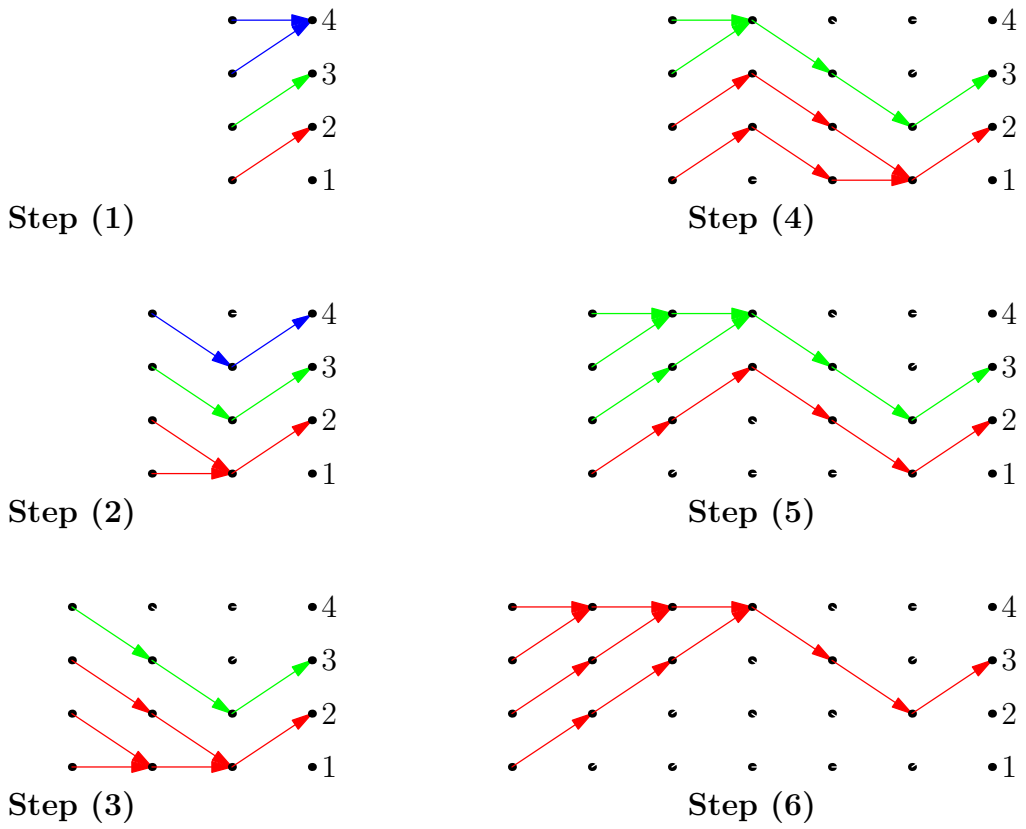

Figure 2: *A run of CFTP procedure, the output is* 3.

This defines a metric space M_2 which turns out to be a complete space. To show the existence and uniqueness of a solution of a stochastic equation we can show that it is a fixed point of a function that is a contraction. For the case of equation 1 we show that a $T : F \mapsto \mathcal{L}\Big((1+X)U\Big)$, with $X \sim F$ and U an independent uniform, is a contraction for d_2. To show this we consider variable X and Y with distributions F and G, we then construct the random variables $(1+X)U$ and $(1+Y)U$ with the same U. These random variables respectively have distributions $T(F)$ and $T(G)$.

$$d_2(T(F), T(G)) \leq ||(X-Y)U||_2 = \sqrt{\frac{1}{3}}||(X-Y)||_2$$

Then by taking the infimum on all possible couplings of X and Y, we get

$$d_2(T(F), T(G)) \leq \frac{1}{\sqrt{3}} d_2(F, G)$$

Then we consider the following sequence $X_{n+1} = (1+X_n)U_{n+1}$, we have $\mathcal{L}(X_{n+1}) = T(\mathcal{L}(X_n))$, the fact that T is a contraction shows that $\mathcal{L}(X_n)$ is a Cauchy sequence in M_2 which is a complete space so $\mathcal{L}(X_n)$ has a limit and it is a fixed point of T. So the distribution we are studying is the limit distribution of the sequence X_n. For more general theorems on the solutions of stochastic equations, see [13].

These distributions that verify stochastic equations often arise in the study of the limit laws, for example in the Quickselect algorithm in [6], [7] and [1] but also as limit distributions in other structures as in [11]. In [2], Devroye and Neininger give properties and methods for approximating the densities of such distributions.

2.7 Problems to face

The straightforward application of coupling from the past for our sequence $X_{n+1} = U_{n+1}(X_n + 1)$ would be to start a chain from every possible $x \in \mathbb{R}_+$ and for the coupling, use the same U_{n+1} for every one of these chains. We see

there are two main problems : we do not have a maximal state and this coupling never gives coalescence, simply because the function $x \mapsto (1+x)u$ is injective.

The first problem is really intrinsic to the CFTP method. In fact applying CFTP requires that the convergence speed of the Markov chain be "the same" for all the starting points, and if we consider our unbounded space, chains beginning at $1,000$ will take longer to approach equilibrium than chains beginning at 1, we say that the chain is not uniformly ergodic. For a precise description of this fact, see [5]. So it is impossible to get complete coalescence of all the chains. Meanwhile one can use the idea of dominated CFTP, introduced by Kendall in [8]. The idea is to use a "simple" Markov chain D coupled to our Markov chain X such that D dominates X. Here simple means that the Markov chain has a stationary distribution that we can sample from, and also that we can simulate the chain backwards. The dominating chain basically says that a chain starting at time $-\infty$ that is under D will always stay under D, and in this way we only have to check the coalescence of chains beginning at a state under the dominating chain.

The second problem can be solved by using more appropriate couplings. For this we have to write the update functions in a different way, or we use ideas like layered multishift coupling as in [16].

First we study the Dickman distribution defined by equation 1, then we try to generalize the method. Then combining ideas from this method and [9], we present an even more general method. Then we consider different cases where the random variable B is unbounded.

3 The procedure

3.1 Introduction

We want to sample from the limit distribution of the sequence of random variables defined by $X_{n+1} = (1+X_n)U_{n+1}$. Instead of using the natural update function $(x,u) \mapsto u(1+x)$, we will use a different coupling, we define the following mapping

$$f : (x,u,v) \mapsto \lfloor u(x+1) \rfloor + \begin{cases} v & \text{if } \lfloor u(x+1) \rfloor \leq \lfloor x \rfloor \\ v(x - \lfloor x \rfloor) & \text{else} \end{cases}$$

Note that this is just another way of coupling the chains for every x, in fact we have

$$\forall x, f(x,U,V) \stackrel{\mathcal{L}}{=} (1+x)W$$

if U and V are independent uniform random variables and W is a uniform random variable. The chains with update f will be called Y. The reason we choose such an update will appear later, it is mainly to obtain coalescence.

3.2 Domination

And now we say

$$f(x,u,v) \leq \lfloor u(\lfloor x \rfloor + 2) \rfloor + v$$

Now let us call

$$g : (n,u) \mapsto \lfloor u(n+2) \rfloor$$

So we can define an update that dominates f as we have $f(x,u,v) \leq g(\lfloor x \rfloor, u) + v$. The idea is to build a dominating chain D_n, having update $D_{n+1} = g(\lfloor D_n \rfloor, U_{n+1}) + V_{n+1}$ so that we do not have to care about chains that are above D_{-n}. We will be denoting the main chains Y having update $Y_{n+1} = f(Y_n, U_n, V_n)$, and $Z_{n+1} = g(Z_n, U_n)$ the integer part of the dominating chain D_{n+1}. Note now that g has a very nice property, in fact if Z is a Poisson random variable, Z verifies

$$Z \stackrel{\mathcal{L}}{=} g(Z,U)$$

where U is a uniform independent of Z.

In fact if Z is a Poisson random variable with parameter 1,

$$\begin{aligned}\mathbb{P}\{g(Z,U)=n\} &= \sum_{k=n-1}^{\infty} \mathbb{P}\{Z=k\}\mathbb{P}\{n \le u(k+2) < n+1\} \\ &= \frac{1}{e}\sum_{k=n-1}^{\infty} \frac{1}{k!}\frac{1}{k+2} \\ &= \frac{1}{e}\sum_{k=n-1}^{\infty} \frac{1}{(k+1)!} - \frac{1}{(k+2)!} \\ &= \mathbb{P}\{Z=n\}\end{aligned}$$

The objective now is to construct a stationary chain Z_{-n} respecting the update that we can extend arbitrarily back into the past, so that if we do not get coalescence we go back one step further. Generating Z_0 is easy, because it is a Poisson random variable, then we must construct Z_{-1} so that when we look at the transition Z_{-1}, Z_0 it follows the wanted transition $g(., U)$.

3.3 Time-reversal of the dominating chain

Now that we have Z_0 we want to calculate Z_{-1}, and U_0 such that

- Z_{-1} is a Poisson random variable
- U_0 knowing Z_{-1} is uniform

$$\mathbb{P}\{Z_{-1}=n|Z_0=k\} = \frac{\mathbb{P}\{Z_0=k|Z_{-1}=n\}\mathbb{P}\{Z_{-1}=n\}}{\mathbb{P}\{Z_0=k\}}$$

$$\mathbb{P}\{Z_{-1}=n|Z_0=k\} = \mathbb{I}_{n\ge k-1}\frac{1}{n+2}\frac{k!}{n!} \tag{2}$$

This equation defines the time reversal of the chain Z. We will be using this distribution to simulate Z backwards. It is useful to see that

$$\mathbb{P}\{Z_{-1}=n|Z_0=k\} = \mathbb{I}_{n\ge k-1}(n+1)\frac{k!}{(n+2)!} = \mathbb{I}_{n\ge k-1}(k!)\Big(\frac{1}{(n+1)!} - \frac{1}{(n+2)!}\Big)$$

Now we can see that by taking

$$U_0 = \frac{Z_0 + U^*}{Z_{-1}+2}$$

with U^* an independent uniform variable, U_0 has the desired distribution.

3.3.1 Coalescence

Note that if the dominating process reaches $D_{-n} < 1$ at step $-n$, then we are assured that all chains using update f beginning at time $-m$ before $-n$ $(m > n)$ will coalesce at time $-n$. This is because the update f is built in such a way that its image has only one point in intervals $[k, k+1)$, so if $D_{-n} < 1$, the paths have no choice ! So the algorithm will consist on waiting until $D_{-n} < 1$ that is $Z_{-n} = 0$, and then we just simulate the chain Y^n that begins at a uniform point in $[0, 1]$ that we called V_{-n}, and that represents the range of the update function when applied to points between 0 and D_{-n-1}.

3.3.2 The algorithm

Let us sum up the algorithm :

- Generate Z_0 as a Poisson random variable
- Update Z backwards until $Z_{-T} = 0$, we have for all $n \leq T$, $Z_{-n+1} = \lfloor U_{-n+1}(Z_{-n} + 2) \rfloor$.
- Now we say $Y^T_{-T} = V_{-T}$, where V_{-T} is a uniform, and update the process Y^T using f, using the **same** uniforms U_{-n} as the ones used for updating Z, and new independent uniform variables V_{-n}
- Return Y_0

An implementation in C of this algorithm can be found in the appendix.

Theorem 1. *The algorithm terminates with probability one, with an expected running number of steps to the past equal to* $1 + \int_0^1 \frac{e^t - 1}{t} dt < 2.32$. *The* Y_0 *returned by the algorithm has the wanted distribution.*

Proof. For the analysis of the running time, see the next section.

We construct the sequence of random variables defined by : X_0 is a uniform, and X_{n+1} is defined by $X_{n+1} = W_{n+1}(X_n + 1)$ with W_n being independent uniforms. Let us define Y^n, by Y^n_{-n} is a uniform and is updated forwards using f with the same U_k's as for the process Z and with independent V_k's (that are the same for all the Y^n's). Then we have $Y^n_0 \stackrel{\mathcal{L}}{=} X_n$ for all n, so Y^n_0 converges in distribution to the Dickman distribution because X_n does so.

Now, consider T such that $Z_{-T} = 0$, and consider Y^n_{-T} for $n \geq T$, i.e. chains that begin before $-T$. Because the process $Z + V$ dominates Y^n, as we mentioned we know we have coalescence at time $-T$, so we have $Y^n_{-T} = Y^T_{-T} = V_{-T}$ for all $n \geq T$. And hence $Y^n_0 = Y^T_0$ because we use the same randomness. And because $T < \infty$ with probability one, the sequence of random variables Y^n_0 converges almost surely to Y^T_0. But we know that Y^n_0 converges in distribution to the Dickman distribution, as a consequence the random variable Y^T_0 has Dickman distribution. □

3.3.3 Analysis of running time

We take as a measure of the running time the number of steps to the past needed. In this way the running time is the time taken by the time reversal of the dominating chain to reach 0. The transition probabilities for this chain are given in equation 2. So let us calculate the expected time T_{n0} to go from state n to state 0, for this, we begin by calculating $\mathbb{E}\{T_{k(k-1)}\}$, the expected time to go from state k to $k-1$.

If we write

$$T_{k(k-1)} = \sum_{i=1}^{N_{up}} T^{\geq k}_{kk,(i)} + 1$$

where N_{up} is the number of excursions made from state k before going to state $k-1$, and $T^{\geq k}_{kk}$ is the length of one such excursion, and because N_{up} and $T^{\geq k}_{kk,(i)}$ are independent we get

$$\mathbb{E}\{T_{k(k-1)}\} = 1 + \mathbb{E}\{N_{up}\}\mathbb{E}\{T^{\geq k}_{kk}\}$$

Remember that when we are at state k with $k \geq 1$, we go to state $k-1$ with probability $\frac{k}{k+1}$ as we can see from equation 2. So

$$\mathbb{E}\{N_{up} + 1\} = \frac{k+1}{k}$$

As a consequence

$$\mathbb{E}\{N_{up}\} = \frac{1}{k}$$

Now we consider $\mathbb{E}\{T^{\geq k}_{kk}\}$. To evaluate this quantity, we consider the chain from which we remove the states $k-1$ and below. The quantity we are looking for is the mean recurrence time for state k in this chain. So the only thing we have to do is get the stationary distribution of this Markov chain. Note that for state k, we remove the

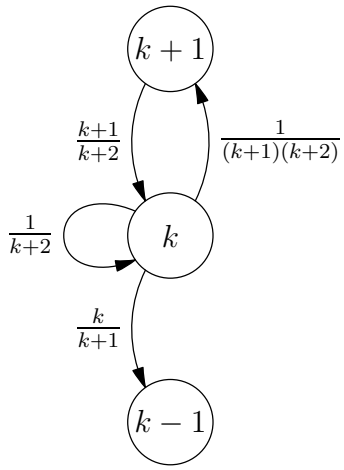

Figure 3: *Some transitions of the time reversal of Z*

transition going to $k-1$, so the probabilities for other transitions departing from k should be divided by $\frac{1}{k+1}$. If $(p_i)_{i\geq k}$ is the stationary probability distribution then we show by induction that $p_{i+1} = \frac{1}{i+1}p_i$ for all $i \geq k$.

In fact

$$p_k = (k+1) \times \frac{1}{k+2}p_k + \frac{k+1}{k+2}p_{k+1}$$

which means

$$p_{k+1} = \frac{1}{k+1}p_k.$$

And for $i \geq k$

$$\begin{aligned}
p_i &= (k+1) \times \frac{1}{k+2}\frac{k!}{i!}p_k + \sum_{j=k+1}^{i} \frac{1}{j+2}\frac{j!}{i!}p_j + \frac{i+1}{i+2}p_{i+1} \\
&= \frac{k+1}{i+2}\frac{k!}{i!}i \times \cdots \times (k+1)p_i + \sum_{j=k+1}^{i} \frac{1}{i+2}\frac{j!}{i!}i \times \cdots \times (j+1)p_i + \frac{i+1}{i+2}p_{i+1} \\
&= \frac{k+1}{i+2}p_i + \sum_{j=k+1}^{i} \frac{1}{i+2}p_i + \frac{i+1}{i+2}p_{i+1} \\
p_{i+1} &= \frac{1}{i+1}p_i.
\end{aligned}$$

Remember what we wanted from this distribution is the mean recurrence time for state k, that is $\frac{1}{p_k}$, we can calculate this by saying that the sum of p_i's is 1, that is

$$\sum_{i=k}^{\infty} \frac{1}{(k+1) \times \cdots \times i}p_k = 1.$$

So

$$\begin{aligned}\frac{1}{p_k} &= k! \sum_{i=k}^{\infty} \frac{1}{i!} \\ &= k! \int_0^1 e^t \frac{(1-t)^{k-1}}{(k-1)!} dt \\ &= k \int_0^1 e^t (1-t)^{k-1} dt\end{aligned}$$

using Taylor's formula. So going back to $\mathbb{E}\{T_{k(k-1)}\}$, we get

$$\begin{aligned}\mathbb{E}\{T_{k(k-1)}\} &= 1 + \frac{1}{k}\frac{1}{p_k} \\ &= 1 + \int_0^1 e^t (1-t)^{k-1} dt\end{aligned}$$

and as a consequence

$$\begin{aligned}\mathbb{E}\{T_{n0}\} &= \sum_{k=1}^{n} \mathbb{E}\{T_{k(k-1)}\} \\ &= \sum_{k=1}^{n} 1 + \int_0^1 e^t (1-t)^{k-1} dt \\ &= n + \int_0^1 e^t \frac{1-(1-t)^n}{t} dt.\end{aligned}$$

Finally we get the expected running time

$$\begin{aligned}\mathbb{E}\{T\} &= \sum_{n=0}^{\infty} \mathbb{P}\{Z=n\}\mathbb{E}\{T_{n0}\} \\ &= \frac{1}{e} \sum_{n=0}^{\infty} \frac{1}{n!} \Big(n + \int_0^1 e^t \frac{1-(1-t)^n}{t} dt \Big) \\ &= 1 + \frac{1}{e} \sum_{n=1}^{\infty} \frac{\int_0^1 e^t \frac{1-(1-t)^n}{t} dt}{n!} \\ &= 1 + \frac{1}{e} \int_0^1 e^t \sum_{n=1}^{\infty} \frac{1-(1-t)^n}{n!t} dt \\ &= 1 + \frac{1}{e} \int_0^1 e^t \frac{1}{t}(e - e^{1-t}) dt \\ &= 1 + \int_0^1 \frac{e^t - 1}{t} dt \\ &< 2.32.\end{aligned}$$

Note that we could also take the number of "elementary" operations as a measure of the running time. In fact at each step we sample from a distribution with infinite tail, and to do this we use the inversion method : we take

a uniform random variable U and get the minimal n such that $\sum_{i=1}^{n} p_i < U$, and thus we have a random time for each step, and the expected time is the expectation of the random variable, more precisely it is $\mathbb{E}\{Z_{-1} - (k-2)\}$, which does not exceed $e-1$. In fact for $k \geq 1$,

$$\begin{aligned}\mathbb{E}\{Z_{-1} - (k-2)\} &= \sum_{j=0}^{\infty} \mathbb{P}\{Z_{-1} > j + k - 2\} \\ &= \sum_{j=0}^{\infty} k! \frac{1}{(j+k)!} \\ &\leq e - 1\end{aligned}$$

We also have to add to this a deterministic time 1 for applying the forward update.

So by using Wald's lemma we can see that the expected number of computational operations performed is 1 for time 0 and the expected number of steps back in the past times the expected number of operations per step which is less that $1 + e^2 < 8.39$. This quantity has actually no real meaning because we did not precisely define what an elementary operation was, but this was just to show that the random computation times for each step are not too large.

Note also that the expected memory used for keeping the uniforms is the expected number of steps taken into the past which is less than 2.32 (this is only the memory for the "reusing randomness" part of CFTP, of course we have to add the memory for current computation).

3.4 Generalization

Let us now consider the distribution that is a solution of

$$X \stackrel{\mathcal{L}}{=} U^{\alpha}(1+X)$$

Note that, in general, we don't have a nice coalescing mapping separating the integer and the fractional part like f because the distribution of $U^{\alpha}(1+x)$ conditioned on $n \leq U^{\alpha}(x+1) < n+1$ depends on n and on x now. But it does not depend on x for $n = 0$, so we choose an update f_{α} that coalesces only on $[0, 1)$

$$f_{\alpha}(x, u, v) = \begin{cases} u^{\alpha}(1+x) & \text{if } u^{\alpha}(1+x) \geq 1 \\ v^{\alpha} & \text{if } u^{\alpha}(1+x) < 1 \end{cases}$$

And we have, if U is a uniform and V, W are independent uniforms, then

$$f_{\alpha}(x, V, W) \stackrel{\mathcal{L}}{=} U^{\alpha}(1+x)$$

3.4.1 Vervaat with $\alpha \geq 1$

Let us start with $\alpha \geq 1$. We also define the sequence $Z_{n+1} = \lfloor U_{n+1}(Z_n + 2) \rfloor$. We still have that $Z_n + 1$ dominates a process Y_n that uses update f_{α}. In fact if $\lfloor Y_n \rfloor \leq Z_n$ then $\lfloor Y_{n+1} \rfloor \leq \lfloor U_{n+1}(\lfloor Y_n \rfloor + 2) \rfloor \leq Z_{n+1}$.

The procedure is the same: we wait until $Z_{-T} = 0$, and then we simulate a chain Y^T starting with Y^T_{-T}, a uniform to the power α, and having update f_{α}. And by using the same arguments, we get that Y^T_0 has the wanted distribution. Because we use the exact same dominating process and the dominating process determines the coalescence, we have the same expected number of steps.

3.4.2 Vervaat with $\alpha < 1$

The problem now is that we have to change the dominating process. The analog of the update g used for the process Z would be here

$$g_{\alpha}(n, u) = \lfloor u^{\alpha}(n+2) \rfloor$$

But now we don't have a simple distribution for a random variable Z that would verify $Z \stackrel{\mathcal{L}}{=} g_\alpha(Z,U)$. What we do is find $\mathbb{P}\{Z=k\}$ as a function of $\mathbb{P}\{Z=0\}$, but calculating the normalization constant is hard so we use rejection sampling techniques to draw from such a distribution.

Let us begin with the calculation of the values of $\mathbb{P}\{Z=k\}$. We can verify by calculation that the probability distribution defined by

$$\mathbb{P}\{Z=k\} = \frac{\prod_{i=1}^{k}(i+1)^\beta - i^\beta}{(k!)^\beta}\mathbb{P}\{Z=0\}$$

with $\beta = \frac{1}{\alpha}$ verifies $Z \stackrel{\mathcal{L}}{=} g_\alpha(Z,U)$. We can determine the distribution with

$$\mathbb{P}\{Z=0\} = \frac{1}{\sum_{k=0}^{\infty}\frac{\prod_{i=1}^{k}(i+1)^\beta - i^\beta}{(k!)^\beta}}$$

but this sum is hard to evaluate. To draw a sample from this distribution, we will bound this sum with an easy sum. We will introduce $p_k = \frac{\prod_{i=1}^{k}(i+1)^\beta - i^\beta}{(k!)^\beta}$, our objective is to draw a sample from the distribution that is proportional to (p_k).

Using a Taylor inequality we get $(i+1)^\beta - i^\beta \leq \beta(i+1)^{\beta-1}$, because $\beta > 1$.

$$\frac{\prod_{i=1}^{k}(i+1)^\beta - i^\beta}{(k!)^\beta} \leq \beta^k \frac{(k+1)^\beta}{k!}$$

Now by introducing $c_\beta = \frac{(\beta/\ln\beta)^\beta}{\beta^{\beta/\ln\beta}}$, a bound of the sequence $\frac{(k+1)^\beta}{\beta^{k+1}}$, we get

$$(k+1)^\beta \leq c_\beta \beta^{k+1}$$

and hence

$$\frac{\prod_{i=1}^{k}(i+1)^\beta - i^\beta}{(k!)^\beta} \leq c_\beta \frac{(\beta^2)^{k+1}}{k!} \stackrel{def}{=} q_k$$

And now this sum can be calculated,

$$\sum_{k=0}^{\infty} q_k = c_\beta \beta^2 e^{\beta^2}$$

The idea, is to draw a uniform between 0 and $c_\beta\beta^2e^{\beta^2} = R$ and to reject the sample if we get something bigger than $S = \sum p_k$. Even though we don't know S, we have a increasing sequence converging to S, $(\sum_{k=0}^{n} p_k)_n$, and also a decreasing sequence $(\sum_{k=0}^{n} p_k + \sum_{k=n+1}^{\infty} q_k)_n$ that we can calculate and that converges to S. So if our sample $U < S$, there will exist a finite n for which $U < \sum_{k=0}^{n} p_k$, and if $U > S$ there will be an n such that $U > \sum_{k=0}^{n} p_k + \sum_{k=n+1}^{\infty} q_k$.

After this, we apply the same idea of taking the time reversal of the chain with update g_α, and we wait till the dominating process reaches 1, i.e. Z reaches 0 to make forwards simulations, using update function f_α. The transition probabilities for calculating Z backwards are

$$\begin{aligned}\mathbb{P}\{Z_{-1}=n|Z_0=k\} &= \frac{\mathbb{P}\{Z_0=k|Z_{-1}=n\}\mathbb{P}\{Z_{-1}=n\}}{\mathbb{P}\{Z_0=k\}} \\ &= \mathbb{I}_{n\geq k-1}\left[\left(\frac{k+1}{n+2}\right)^\beta - \left(\frac{k}{n+2}\right)^\beta\right]\frac{\prod_{i=1}^{n}(i+1)^\beta - i^\beta}{\prod_{i=1}^{k}(i+1)^\beta - i^\beta}\frac{(k!)^\beta}{(n!)^\beta}\end{aligned}$$

Note here that determining the innovations U_{-n} from Z_{-n} and Z_{-n-1} is not as straightforward as in the case $\alpha = 1$, so we just generate independent uniform variables and we take U_{-n} to be the first one such that $Z_{-n} = \lfloor U_{-n+1}^{\alpha}(Z_{-n-1}+2)\rfloor$.

This generalization is not practical at all, in fact even for $\alpha = 0.1$, it takes more than a minute to get one sample, compared to more than 50,000 samples for $\alpha = 1$ for the same time. We now introduce another method which solves more general cases and is significantly faster for small values of α.

4 A general method

We now present a general method for $X \stackrel{\mathcal{L}}{=} AX + B$ with $0 \leq A \leq 1$ and with B positive and bounded $B \leq c$. We show that this case comes down to finding a method for $X \stackrel{\mathcal{L}}{=} A(X+1)$. First consider $X \stackrel{\mathcal{L}}{=} AX + c$, by introducing $Y = X - c$ we get $Y = A(Y + c)$ and $Z = Y/c$ we get $Z = A(Z + 1)$. So we can just simulate Z and then take $X = cZ + c$. But if we can get a dominating chain to sample from the solution of $X \stackrel{\mathcal{L}}{=} AX + c$, then using the exact same domination we can sample from $X \stackrel{\mathcal{L}}{=} AX + B$.

Now let us consider $X \stackrel{\mathcal{L}}{=} A(X + 1)$, and the sequence $X_{n+1} = A_{n+1}(X_n + 1)$. Note that $\lfloor X_{n+1} \rfloor$ can be either $\lfloor X_n \rfloor + 1$, $\lfloor X_n \rfloor$ or less than $\lfloor X_n \rfloor$. These three events will be what drives our dominating process. Let us calculate the probability of these events, for a fixed x, we have

$$\mathbb{P}\{\lfloor A(x+1) \rfloor = \lfloor x \rfloor + 1\} = \mathbb{P}\{A(x+1) \geq \lfloor x \rfloor + 1\} = \mathbb{P}\{A \geq \frac{\lfloor x \rfloor + 1}{x+1}\}$$

Recall the objective is to obtain a dominating process, so we want the dominating process to go up with more probability. For this we can have a lower bound on the value of $\frac{\lfloor x \rfloor + 1}{x+1}$, by supposing $x \geq \kappa$ for some positive integer κ. This idea of separating the cases $x \geq \kappa$ and $x \leq \kappa$ is due to Kendall and Thönnes in [9].

$$x \geq \kappa, \frac{\lfloor x \rfloor + 1}{x+1} \geq 1 - \frac{1}{\kappa + 1}$$

Now the event $UP = \{A \geq 1 - \frac{1}{\kappa+1}\}$ has a probability greater than event $\lfloor A(x+1) \rfloor = \lfloor x \rfloor + 1$, for $x \geq \kappa$. We also define the events $EQ = \{1 - \frac{2}{\kappa+1} \leq A \leq 1 - \frac{1}{\kappa+1}\}$, and $DOWN = \{A \leq 1 - \frac{2}{\kappa+1}\}$.

Now let us define the following update for our dominating process:

$$\phi(n, A) = \begin{cases} n+1 & \text{if } 1 - \frac{1}{\kappa+1} \leq A \leq 1 \\ n & \text{if } 1 - \frac{2}{\kappa+1} \leq A \leq 1 - \frac{1}{\kappa+1} \\ \max(\kappa, n-1) & \text{if } 0 \leq A \leq 1 - \frac{2}{\kappa+1} \end{cases}$$

So the sequence defined by $Z_{n+1} = \phi(Z_n, A_{n+1})$ is a candidate dominating process for $\lfloor X_n \rfloor$, and so, $D_n = Z_n + 1$ would dominate the sequence X_n.

Now Z has some nice properties:

- Simple random walk on $\mathbb{Z}_+$.
- If we choose κ big enough such that $\mathbb{P}\{DOWN\} > \mathbb{P}\{UP\}$ then the chain has limit distribution $\kappa + Geom(\frac{\mathbb{P}\{UP\}}{\mathbb{P}\{DOWN\}})$. In fact $\mathbb{P}\{Z = \kappa + n\} = (1 - \frac{\mathbb{P}\{UP\}}{\mathbb{P}\{DOWN\}})\frac{\mathbb{P}\{UP\}}{\mathbb{P}\{DOWN\}}^n$.
- Z is time reversible, this means that $\mathbb{P}\{Z_{-1} = n | Z_0 = k\} = \mathbb{P}\{Z_0 = n | Z_{-1} = k\}$, so it is really easy to simulate backwards.

For Vervaat perpetuities, we can check that taking $\kappa = \lfloor \frac{2}{1-(1/2)^\alpha} \rfloor$ works fine.

It remains to find a good way of coupling the chains so that coalescence can happen. In fact, if we apply the trivial coupling of applying the same A for all chains, we will never get the chains to couple. But note that it is important that the coupling we choose be monotone, i.e. if $x \leq y$ then $N(x) \leq N(y)$ where N is our random update function.

We use multishift coupling as described in [16]. We will introduce a monotone coupling but the image of the set of all the states lying in an interval will be discrete. Let us first consider a simple case, we want to construct a coupling for the transition $\phi : s \mapsto s + V_s$, where V_s is a uniform for every s. By generating a uniform V_0 and considering the mapping $\phi : s \mapsto \lceil s - V_0 \rceil + V_0$, it verifies the wanted property that is for all s, $\phi(s) - s$ has a uniform distribution. The idea is that we build a line grid with length 1, and the image of our mapping must lie in one of the points in $\mathbb{Z} + U_0$. So each point s is mapped to the first point in this grid that is greater than s.

Now let us complicate things a little bit more and suppose V has a general distribution having a density function f. For simplicity we suppose that f is unimodal, that is there is an m such that f increasing for $x \leq m$ and decreasing for $x \geq m$. To do this, we observe that : if (X, Y) is a uniform point on the surface $S_f = \{(x, y) \in \mathbb{R}^2 | 0 \leq y \leq f(x)\}$, and X_1 and X_2 is such that $f(X_1) = f(X_2) = Y$, then choosing a Z uniformly on $[X_1, X_2]$ yields another point

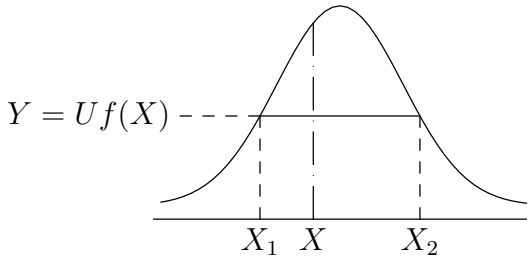

Figure 4: *Layered multishift coupling*

(Z, Y) that is also distributed uniformly on S_f. In this way, we get back to the case of V being a uniform. Here is how to do it.

- Choose X having density f.
- Let $Y = Uf(X)$ with U a uniform, so that (X, Y) is uniform on S_f.
- Calculate X_1 and X_2, such that $f(X_1) = f(X_2) = Y$, $X_1 < X_2$.
- The unit of the grid is now $L = X_2 - X_1$, the range of the mapping is $L\mathbb{Z} + (X - X_1)$.
- Finally $\phi(s) = \min\{x \in L\mathbb{Z} + (X - X_1), x \geq s\}$.

Remember our update is $\phi : s \mapsto A(s+1) = e^{\ln A + \ln(s+1)}$. We know we can sample from $\psi : s \mapsto \ln(s+1) + \ln A$ provided A has a density that we know. Then we take $\phi(s) = e^{\psi(s)}$.

Suppose we are at time $-n$, we have a method that generates one random variable $\ln A$ and use it for all the chains. And because our coupling is monotone, we can only simulate one lower chain and one upper chain. The lower chain will begin at state 0 and the upper chain at state D_{-n}. The problem is that when simulating the dominating chain, we already produced the random variable A_{-n+1} and we have to use the same one so that our domination argument remains valid. In fact, when we make the calculations, we suppose the randomness is the same for both chains. But if we use the same A_{-n+1} for all the chains we will never get coalescence.

To solve this problem, notice that the coupling we defined earlier has some kind of origin at point 0 (as an origin for the grid). But we can choose this origin arbitrarily. This means we are free to choose one point and define its image. Remember the important thing is to preserve the domination of the dominating chain. So if, for the upper chain, we use the random variable A_{-n+1} for the update so that D_{-n} gets mapped to $A_{-n+1}(D_{-n} + 1)$ and depending on this we create the grid described by multishift coupling, we will know the state 0 gets mapped to a state lower that $A_{-n+1}(D_{-n} + 1)$ and thus is dominated by Z_{-n+1}.

Let us sum up the procedure by giving an overview, we suppose we have a valid κ, such that $\mathbb{P}\{DOWN\} > \mathbb{P}\{UP\}$:

1. Generate Z_0 with distribution $\mathbb{P}\{Z = \kappa + n\} = (1 - \frac{\mathbb{P}\{UP\}}{\mathbb{P}\{DOWN\}})\frac{\mathbb{P}\{UP\}}{\mathbb{P}\{DOWN\}}^n$.
2. Generate a random variable A_0' with the distribution of A and let $Z_{-1} = \phi(Z_0, A_0')$.
3. Generate A_0 such that $Z_0 = \phi(Z_{-1}, A)$ (we can do this by generating A_0's until getting a valid one).
4. Let $Y_{-1}^{up,-1} = Z_{-1} + 1$, and $Y_0^{up,-1} = A_0(Y_{-1}^{up,-1} + 1)$.
5. Let $Y_{-1}^{low,-1} = 0$, and $Y_0^{low,-1}$ is calculated using multishift coupling.
6. If we get $Y_0^{up,-1} = Y_0^{low,-1}$ then we return this value, else we go back one step with Z, and then do simulations forward.

Note that we must keep the randomness used at each step so that the mapping for each time be always the same. This applies also for the random variables involved in the multishift coupling. We do not give the detailed steps of the algorithms to avoid introducing too much notations.

What remains to show is that coalescence happens with probability 1. For this, we try to bound the expected time until coalescence. We give a sketch of a proof that the expected time is finite in the case of perpetuities, i.e. $A = U^\alpha$. For this suppose that the dominating chain has value $\kappa+1$, then the probability that the upper chain and the lower chain coalesce in one step is a positive constant $p > 0$, in fact if the grid unit L is big enough, $\kappa + 1$ and 0 get mapped to the same value. This does not finish the proof since the dominating chain is not always at $\kappa + 1$. But the state $\kappa+1$ of the chain has mean recurrence time $\frac{1}{(1-\frac{\mathbb{P}\{UP\}}{\mathbb{P}\{DOWN\}})}$. At each time the dominating chain has value $\kappa + 1$, we have a probability at least p of coalescence. So, as these events at different steps are independent, the expected time in the past until coalescence is finite.

This method has been implemented in MATLAB and tested on Vervaat perpetuities and works roughly as fast as the method presented in section 3 for $\alpha = 1$ but significantly faster for $\alpha < 1$.

5 Other types of perpetuities

We have not yet considered the case where one of the two random variable A or B is unbounded. We consider the following equation

$$X \stackrel{\mathcal{L}}{=} qX + B \tag{3}$$

with $0 < q < 1$ and B a positive random variable.

Let us first consider the existence of such a distribution. If we suppose that $\mathbb{E}\{B^2\} < \infty$, the function $T : F \mapsto \mathcal{L}(qX + B)$ with $X \sim F$ is well defined from M_2 into itself. Moreover for two random variables X and Y having distribution functions F and G, by tacking random variables with distributions $T(F)$ and $T(G)$ with the same value B, we get

$$d_2(T(F), T(G)) \leq (\mathbb{E}\{(q(X-Y))^2\})^{1/2} = q||(X-Y)||_2$$

And by taking the infimum on all the possible couplings of X and Y, we get that T is a contraction and hence the existence and uniqueness of such a distribution. We can also construct a sequence that converges in law to this distribution as follows

$$X_{n+1} = qX_n + B_n$$

with B_n a sequence of independent random variables distributed like B, in other words,

$$X_n = \sum_{k=1}^{n} q^k B_k.$$

The objective is to simulate this distribution perfectly in finite time. As usual, for this, we introduce a dominating chain. Note that if B is bounded by M, then we can define a dominating chain to be the constant $\frac{1}{1-q}M$. In the general case we take

$$Z_{n+1} = \max\Big(\frac{B_n}{1-\sqrt{q}}, \sqrt{q}Z_n\Big).$$

We can easily verify that the mapping for a fixed B, $f : x \mapsto qx + B$ is dominated by the mapping $g : x \mapsto \max(\frac{B}{1-\sqrt{q}}, \sqrt{q}x)$. We need to look now for the stationary distribution of this chain. We can write it as follows

$$Z_\infty = \frac{1}{1-\sqrt{q}} \max_{n\geq 0} \sqrt{q}^n B'_n$$

with B'_n distributed as B. We see immediately that $Z_\infty \stackrel{\mathcal{L}}{=} \max\Big(\frac{B}{1-\sqrt{q}}, \sqrt{q}Z_\infty\Big)$.

5.1 Simulation of the stationary distribution of the dominating chain

We will use rejection sampling method, by finding an upper bound of the density function. We write the distribution function F of Z_∞, as a function of the distribution function G of B. We make the full calculations only for B having exponential distribution.

$$F(x) = \mathbb{P}\{Z_\infty \leq x\} = \prod_{n=0}^{\infty} \mathbb{P}\{\sqrt{q}^n B'_n \leq x\} = \prod_{n=0}^{\infty} G(\frac{x}{\sqrt{q}^n})$$

Thus, the density function can be written if we denote by g the density function of B

$$f_{Z_\infty}(x) = \sum_{k=0}^{\infty} \frac{1}{\sqrt{q}^k} \frac{g(\frac{x}{\sqrt{q}^k})}{G(\frac{x}{\sqrt{q}^k})} F(x)$$

Lemma 1 (Dominating the density)**.** *We define $\rho : x \mapsto \frac{xg(x)}{G(x)}$, if ρ is a decreasing function, and the distribution function F of Z_∞ verifies $F(x) \leq x$ then we have the bound*

$$f_{Z_\infty}(x) \leq \rho(x) - \frac{1}{1-\sqrt{q}} \ln(G(x))$$

and when B is exponential, the conditions are met and we get

$$f_{Z_\infty}(x) \leq \frac{xe^{-x}}{1-e^{-x}} - \frac{1}{1-\sqrt{q}} \ln(1-e^{-x}).$$

Proof. If we suppose $F(x) \leq x$ then,

$$f_{Z_\infty}(x) \leq \sum_{k=0}^{\infty} \frac{x}{\sqrt{q}^k} \frac{g(\frac{x}{\sqrt{q}^k})}{G(\frac{x}{\sqrt{q}^k})}$$

then by definition of ρ,

$$f_{Z_\infty}(x) \leq \sum_{k=0}^{\infty} \rho(\frac{x}{\sqrt{q}^k}).$$

Now because ρ is decreasing, we have

$$f_{Z_\infty}(x) \leq \sum_{k=0}^{\infty} \rho(\frac{x}{\sqrt{q}^k})$$

and we are going to bound this function by an integral, because ρ is decreasing, we have

$$\rho(\frac{x}{\sqrt{q}^{k+1}}) \leq \frac{1}{\frac{x}{\sqrt{q}^{k+1}} - \frac{x}{\sqrt{q}^k}} \int_{\frac{x}{\sqrt{q}^k}}^{\frac{x}{\sqrt{q}^{k+1}}} \rho(y)dy$$

but we also have

$$\frac{1}{\frac{x}{\sqrt{q}^{k+1}}} \rho(\frac{x}{\sqrt{q}^{k+1}}) \leq \frac{1}{\frac{x}{\sqrt{q}^{k+1}} - \frac{x}{\sqrt{q}^k}} \int_{\frac{x}{\sqrt{q}^k}}^{\frac{x}{\sqrt{q}^{k+1}}} \frac{\rho(y)}{y} dy.$$

And thus

$$\frac{x}{\sqrt{q}^{k+1}} \frac{1}{\frac{x}{\sqrt{q}^{k+1}}} \rho(\frac{x}{\sqrt{q}^{k+1}}) \leq \frac{1}{1-\sqrt{q}} \int_{\frac{x}{\sqrt{q}^k}}^{\frac{x}{\sqrt{q}^{k+1}}} \frac{\rho(y)}{y} dy.$$

As a consequence

$$f_{Z_\infty}(x) \leq \rho(x) + \frac{1}{1-\sqrt{q}} \int_x^{\infty} \frac{\rho(y)}{y} dy = \rho(x) - \frac{1}{1-\sqrt{q}} \ln(G(x))$$

In the exponential case we have :

- $F(x) \leq 1 - e^{-x} \leq x$
- We see that $\rho(x) = \frac{xe^{-x}}{1-e^{-x}}$ is decreasing by calculating the derivative

Hence, we can apply the bound

$$f_{Z_\infty}(x) \leq \frac{xe^{-x}}{1-e^{-x}} - \frac{1}{1-\sqrt{q}} \ln(1-e^{-x})$$

□

For the rest, we will only consider the case where B is exponential. We need to do some more work to get to simple densities that we can simulate. Near 0 it is $\ln(1-e^{-x})$ that is important, we bound it for $x \leq 1$ as follows

$$\ln\left(\frac{1}{1-e^{-x}}\right) = \ln\left(\frac{1}{x}\frac{x}{1-e^{-x}}\right)$$

but for $x \leq 1$, $\frac{x}{1-e^{-x}} \leq \frac{1}{1-e^{-1}}$ and thus

$$\ln\left(\frac{1}{1-e^{-x}}\right) \leq \ln\left(\frac{1}{x}\right) + \ln\left(\frac{e}{e-1}\right).$$

And for $x \geq 1$, we bound the term $\frac{1}{1-e^{-x}} \leq \frac{1}{1-e^{-1}}$, and we write $\ln\left(\frac{1}{1-e^{-x}}\right) = \ln\left(1 + \frac{e^{-x}}{1-e^{-x}}\right) \leq \frac{e^{-x}}{1-e^{-x}}$ so we get

Lemma 2 (Domination of the density f_{Z_∞} with a weighted sum of easy densities)**.** *If B has exponential distribution, then*

$$f_{Z_\infty}(x) \leq \begin{cases} \frac{e}{e-1}e^{-x} + \frac{1}{1-\sqrt{q}}\ln(\frac{e}{e-1}) + \frac{1}{1-\sqrt{q}}\ln(\frac{1}{x}) & \text{for } x \leq 1 \\ \frac{e}{e-1}xe^{-x} + \frac{1}{1-\sqrt{q}}\frac{e}{e-1}e^{-x} & \text{for } x > 1 \end{cases}$$

This bound is convenient because each term is the density function of a well-known random variable. So to draw a sample uniformly under the surface defined by this function, we first determine from which one of the five components we will get our sample and then simulate according to it.

The five components can be obtained as follows :

- exponential E with weight $\frac{e}{e-1}$ (we group with part of $x > 1$)
- uniform U with weight $\frac{1}{1-\sqrt{q}}\ln(\frac{e}{e-1})$
- product of independent uniforms $U \times U'$ with weight $\frac{1}{1-\sqrt{q}}$
- shifted exponential $1 + E$ weight $\frac{e}{e-1}(\frac{1}{1-\sqrt{q}} - 1)$
- sum of independent exponentials $E + E'$ restricted to $[1, \infty)$ with weight $\frac{e}{e-1}\frac{2}{e}$

It remains to approximate the density of the target distribution, to decide if we reject or accept a sample from the dominating density.

Remember

$$f_{Z_\infty}(x) = \sum_{k=0}^{\infty} \rho\left(\frac{x}{\sqrt{q}^k}\right)\frac{F(x)}{x}$$

Lemma 3. *For an exponentially distributed B, there exists functions g_n and h_n that we can calculate, such that both $g_n(x)$ and $h_n(x)$ tend to $f_{Z_\infty}(x)$ when n goes to ∞. Moreover for all n, $g_n(x) \leq f_{Z_\infty}(x) \leq h_n(x)$.*

Proof. First we define the partial product $F_n(x) = \prod_{j=0}^{n}(1 - e^{\frac{-x}{\sqrt{q}^j}})$, we have $F_n(x)$ tends to $F(x)$ from above, now let us bound the error :

$$F_n(x) - F(x) = 1 - \prod_{j=n+1}^{\infty}(1 - e^{\frac{-x}{\sqrt{q}^j}}).$$

So we consider the product

$$\prod_{j=n+1}^{\infty}(1 - e^{\frac{-x}{\sqrt{q}^j}}) \geq 1 - \sum_{j=n+1}^{\infty} e^{\frac{-x}{\sqrt{q}^j}}.$$

As a consequence we get

$$F_n(x) - F(x) \leq \frac{1}{1-\sqrt{q}} \frac{1}{x/\sqrt{q}^n} e^{-x/\sqrt{q}^n}$$

so

$$g_n(x) = \sum_{k=0}^{n} \frac{e^{-\frac{x}{\sqrt{q}^k}}}{\sqrt{q}^k(1 - e^{-\frac{x}{\sqrt{q}^k}})} \left(F_n(x) - \frac{1}{1-\sqrt{q}} \frac{\sqrt{q}^n}{x} e^{-\frac{x}{\sqrt{q}^n}}\right) \leq f_{Z_\infty}(x).$$

Now for the upper sequence we have to evaluate,

$$\begin{aligned}
\sum_{k=n+1}^{\infty} \frac{e^{-\frac{x}{\sqrt{q}^k}}}{\sqrt{q}^k(1 - e^{-\frac{x}{\sqrt{q}^k}})} &= \sum_{k=n+1}^{\infty} \frac{e^{-\frac{x}{\sqrt{q}^k}}}{\sqrt{q}^k(1 - e^{-\frac{x}{\sqrt{q}^k}})} \\
&\leq \sum_{k=n+1}^{\infty} \frac{e^{-\frac{x}{\sqrt{q}^k}}}{\sqrt{q}^k \frac{x}{\sqrt{q}^k}} \\
&= \frac{1}{x} \sum_{k=n+1}^{\infty} e^{-\frac{x}{\sqrt{q}^k}} \\
&\leq \frac{1}{x} \sum_{k=n+1}^{\infty} e^{-x \ln(\frac{1}{\sqrt{q}})k} \\
&= \frac{1}{x} e^{x \ln \sqrt{q}(n+1)} \frac{1}{1 - e^{x \ln \sqrt{q}}}
\end{aligned}$$

So if we take

$$h_n = \left(\sum_{k=0}^{n} \frac{e^{-\frac{x}{\sqrt{q}^k}}}{\sqrt{q}^k(1 - e^{-\frac{x}{\sqrt{q}^k}})} + \frac{e^{(n+1)x \ln \sqrt{q}}}{x(1 - e^{x \ln \sqrt{q}})} \right) F_n(x)$$

we get $f_{Z_\infty}(x) \leq h_n(x)$.

□

To sum up the procedure for sampling from Z_∞, we first get a uniform sample on the surface under the function dominating the density f_{Z_∞} defined in lemma 2 by choosing one of the five components, each one with probability proportional to its weight, and then sampling from the chosen component. After this, we have to decide whether this sample point (x, y) should be rejected or not, this is done by calculating the two approximations of $f_{Z_\infty}(x)$, $g_n(x)$ and $h_n(x)$ for increasing values of n. If for some n, we reach $y \leq g_n(x)$, then we can accept the sample x, and if for some n, we get $h_n(x) < y$, the sample x is rejected and we restart the whole process.

5.2 Simulating Z backwards

Remember that we need to be able to simulate the dominating chain backwards, so we have a value for Z_{n+1} and we need to compute Z_n that is well distributed. Recall the update for Z is

$$Z_{n+1} = \max\Big(\frac{B_n}{1-\sqrt{q}}, \sqrt{q}Z_n\Big).$$

To calculate Z_n, we first calculate the probability that $Z_{n+1} = \sqrt{q}Z_n$,

$$\begin{aligned}
\mathbb{P}\{Z_{n+1} = \sqrt{q}Z_n | Z_{n+1} \in [z, z+dz]\} &= \frac{\mathbb{P}\{Z_{n+1} = \sqrt{q}Z_n, Z_{n+1} \in [z, z+dz]\}}{\mathbb{P}\{Z_{n+1} \in [z, z+dz]\}} \\
&= \frac{\mathbb{P}\{\sqrt{q}Z_n \in [z, z+dz], Z_{n+1} = \sqrt{q}Z_n\}}{\mathbb{P}\{Z_\infty \in [z, z+dz]\}} \\
&= \frac{\mathbb{P}\{\sqrt{q}Z_n \in [z, z+dz], \frac{B_n}{1-\sqrt{q}} \leq \sqrt{q}Z_n\}}{\sqrt{q}f_{Z_\infty}(z)dz} \\
&= \frac{\mathbb{P}\{\sqrt{q}Z_n \in [z, z+dz], \frac{B_n}{1-\sqrt{q}} \leq z\}}{\sqrt{q}f_{Z_\infty}(z)dz} \\
&= \frac{f_{Z_\infty}(\frac{z}{\sqrt{q}})}{\sqrt{q}f_{Z_\infty}(z)} G\Big((1-\sqrt{q})z\Big) \\
&= p
\end{aligned}$$

So suppose we can toss a coin with heads having this probability, then, knowing the value z of Z_{n+1} we toss the coin, if we get heads we just assign $Z_n = \frac{z}{\sqrt{q}}$ and we assign B_n to be an exponential restricted to $[0, z]$. In the other case, we know that $Z_{n+1} = \frac{B_n}{1-\sqrt{q}}$, so we assign $B_n = z(1-\sqrt{q})$, and for Z_n, we draw a sample distributed like Z_∞ restricted to $[0, \frac{z}{\sqrt{q}}]$.

It remains to show how to toss a coin with this probability. We simply generate a uniform random variable U on $[0, 1]$ and we have sequences that converge p from above and from below (using functions g_n and h_n introduced in the previous section), so we can decide if $U < p$ or not, and hence make our decision.

5.3 Coupling

The dominating process is now all set up, we can simulate it in the past as far as we want. We have already computed the random variables when simulating the dominating chain. Now we use this randomness for the forward chain X defined by $X_{n+1} = qX_n + B_n$. We actually simulate two chains, the lower chain beginning at 0 and the other beginning at Z_{-n}. We couple this chain using multishift coupling as described in section 4. The global algorithm is basically the same, except that here the dominating chain is more complicated.

Acknowledgments

I would like to thank Luc Devroye, my supervisor, for all explanations and discussions. Also many thanks to the Computational Geometry group at McGill university for hosting me during my training period.

References

[1] Wei-Mei Chen and Hsien-Kuei Hwang. Analysis in distribution of two randomized algorithms for finding the maximum in a broadcast communication model. *Journal of Algorithms, 46:2 (2003), 140-177*, 2003.

[2] L. Devroye and R. Neininger. Density approximation and exact simulation of random variables that are solutions of fixed-point equations. *Advances of Applied Probability, vol. 34, pp. 441-468*, 2002.

[3] Luc Devroye. Simulating perpetuities. *Methodologies and Computing in Applied Probability, vol. 3, pp. 97-115*, 2001.

[4] James Allen Fill and Mark Huber. Personal communication, 2007.

[5] S.G. Foss and R.L. Tweedie. Perfect simulation and backward coupling. *Stochastic Models, 14(1-2):187–203*, 1998.

[6] R. Grübel and U. Rösler. Asymptotic distribution theory for hoare's selection algorithm. *Advances in Applied Probability 28, 252-269*, 1996.

[7] Hsien-Kuei Hwang and Tsung-Hsi Tsai. Quickselect and the dickman function. *Combinatorics, Probability and Computing 11, 353-371*, 2002.

[8] Wilfrid Kendall. Perfect simulation for the area-interaction point process. *Probability Towards 2000, edited by L. Accardi and C.C. Heyde. Springer-Verlag. 218-234*, 1998.

[9] Wilfrid Kendall and Elke Thönnes. Random walk cftp. *unpublished*, 2004.

[10] D.J. Murdoch and P.J. Green. Exact sampling from a continous space. *Scandinavian Journal of Statistics 25(3):483–502*, 1998.

[11] Mathew D. Penrose and Andrew R. Wade. Limit theory for the random on-line nearest-neighbour graph. *arXiv:math/0603561v1*, 2006.

[12] James G. Propp and David B. Wilson. Exact sampling with coupled markov chains and applications to statistical mechanics. *Random Structures and Algorithms, 9(1&2):223–252*, 1996.

[13] Uwe Rösler. A fixed point theorem for distributions. *Stochastic Process. Appl. 42, 195-214*, 1992.

[14] W. Vervaat. On a stochastic difference equation and a representation of nonnegative infinitely divisible random variables. *Adv. Appl. Probab., 11, 750-783*, 1997.

[15] David B. Wilson. Perfectly random sampling with markov chains. http://dbwilson.com/exact/.

[16] David B. Wilson. Layered multishift coupling for use in perfect sampling algorithms. *In Neil Madras, editor, Monte Carlo Methods, volume 26 of Fields Institute Communications, pages 143–179. American Mathematical Society*, 2000.

Code for sampling from Dickman distribution

This is an implementation of the algorithm for sampling from a Dickman distribution. The uniforms used for the updates are stored in a stack while we are going backwards, then when the chain Z reaches 0, we know we have coalescence so we use these uniforms for our forward updates.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/* Returns a uniform on [0,1] */
double rand_uniform() {
    return ((double) rand())/((double) RAND_MAX + 1);
}

/* Returns an integer with Poisson distribution */
int rand_poisson() {
    double u = rand_uniform();
    int n = 0, t = 1;
```

```
    double p = 1;
    for(n=0; ;n++) {
        if(u*exp(1) <= p) {
            return n;
        }
        p = p + 1.0/( (double) t*(n+1));
        t = t*(n+1);
    }
}

/* Stack for reusing the u's */
struct node {
    double data;
    struct node *next;
};

struct node *top;

void push(double y) {
    struct node *x;
    x=malloc(sizeof(struct node));
    x->data = y;
    x->next = top;
    top = x;
}

double pop() {
    double x;
    struct node *temp;
    if(top==NULL) {
        printf("error, stack empty");
        return 0;
    }
    else {
        x=top->data;
        temp = top->next;
        free(top);
        top=temp;
        return x;
    }
}


/* Returns a double with Dickman distribution */
double rand_dickman() {
    // some variables
    int z, new_z;
    double u;
    top = NULL; // the stack holds the u's
    double y;
    int n=0;
    int k, a;
    int N;
    double F;
```

```
    double v, p;
    z = rand_poisson();

    while(1) {
        if(z == 0) {
            y = rand_uniform();
            for(k=(n-1); k >= 0 ; k--) {
                u = pop();
                N = floor(y);
                F = y - N;
                if(floor(u*(N+1+F)) > N) {
                    y = floor(u*(N+1+F)) + F*rand_uniform();
                }
                else {
                    y = floor(u*(N+1+F)) + rand_uniform();
                }
            }
            return y;
        }
        // applying backward update
        v = rand_uniform();
        p = 1/((double) z+1); a = z-1;
        for(k=0 ; k<100;k++) {
            if( 1-p > v ) {
                new_z = a;
                break;
            }
            a = a+1; p = p/(a+2);
        }
        push((z + rand_uniform())/(new_z+2));
        z = new_z;
        n = n+1;
    }
}

int main() {
    srand(1);
    r = rand_dickman();
}
```