Fast Perfect Simulation of Vervaat Perpetuities

Kirkwood Cloud

Pomona College

Mark Huber[*]

Claremont McKenna College

Abstract

This work presents a new method of simulating exactly from a distribution known as a Vervaat perpetuity. This type of perpetuity is indexed by a parameter β. The new method has a bound on the expected run time which is polynomial in β (as β goes to infinity). This is much faster than the previously best known bound due to an earlier method of Fill and the second author, which ran in expected time $\exp(\beta \ln(\beta) + \Theta(\beta))$ as $\beta \to \infty$. The earlier method utilized dominated coupling from the past to place bounds on a stochastic process for perpetuities from above. By extending to an update function that changes based on the dominating process, it is possible to create a new method that bounds the perpetuities from both above and below. This new approach is shown to run in expected time $O(\beta \ln(\beta))$ as $\beta \to \infty$.

Keywords: exact simulation, dominated coupling from the past
2010 MSC: 68Q25, 65C05

1. Introduction

A perpetuity randomly discounts random investments over time. Such random variables arise in economics ([12, 13]), physics ([16]), number theory ([4]), computer science ([14]), and many other fields.

Definition 1. Let $\{A_i\}$ and $\{W_i\}$ be two sequences of independent, identically distributed (iid) random variables. Then a *perpetuity* is a random variable of the form

$$Y = A_1 W_1 + A_2 W_1 W_2 + A_3 W_1 W_2 W_3 + \cdots , \tag{1}$$

Suppose that for all i, W_i has the same distribution as W (write $W_i \sim W$) and $A_i \sim A$. Then it also holds that $Y \sim W(A + Y)$ for Y and W independent.

[*] Corresponding author

Email addresses: kacloud@gmail.com (Kirkwood Cloud), mhuber@cmc.edu (Mark Huber)

Throughout this work it will be assumed that A and W are nonnegative with probability 1, $\mathbb{E}[A] < \infty$, and $\mathbb{E}[W] < 1$. These two assumptions give that Y is nonnegative and finite with probability 1. In fact, the Monotone convergence theorem gives that

$$\mathbb{E}[Y] = \frac{\mathbb{E}[W]\mathbb{E}[A]}{1 - \mathbb{E}[W]}. \tag{2}$$

The interest here is in developing a method for sampling from perpetuities with $\mathbb{E}[W]$ close to 1, so that $\mathbb{E}[Y]$ is very large.

1.1. Vervaat perpetuities

Vervaat [17] studied the convergence properties of perpetuities, and in particular looked at a simple family indexed by a single parameter β. Let $U \sim \mathsf{Unif}([0,1])$ mean that U has the uniform distribution over $[0,1]$.

Definition 2. A *Vervaat perpetuity* is a perpetuity where $W_i \sim U^{1/\beta}$ for some $\beta \in (0,\infty)$ where $U \sim \mathsf{Unif}([0,1])$ and the A_i are identically 1.

For a Vervaat perpetuity Y, $\mathbb{E}[Y] = \beta$, thus giving rise to a simple family where $\mathbb{E}[Y]$ can be directly controlled.

When $\beta = 1$, the classical Dickman [4] distribution is recovered. No closed form for this distribution is known. Originally used to study largest prime factors, this distribution also arises in longest cycles in permutations. See [9] for more details.

In addition, the expected running time of the classic `Quickselect` algorithm of Hoare [8] for finding order statistics of an unsorted set of elements approaches a perpetuity. If one pivot is chosen uniformly, then asymptotically the running time approaches the Dickman distribution (see [14] and [6, Section 1.2]).

(Note that some authors define the Dickman distribution as $1 + Y$ for Y a Vervaat perpetuity with $\beta = 1$.) The first author to simulate exactly from the Vervaat distribution in finite time was Devroye [2] based on envelope refinement of acceptance/rejection. In [2], it was shown that his algorithm ran in finite time for all β, and that for $\beta > 5/3$, the expected running time of the algorithm was finite. However, bounds on the expected running time in terms of β were not given.

As noted in [2, p. 98] by Devroye, Fill had already developed a dominated coupling from the past for perfect simulation from the Dickman distribution. This was extended and analyzed by the second author, eventually resulting in the method in [6] that applied to Vervaat perpetuities for all values of β. In this work, it was shown that the expected number of uniforms N required by the algorithm always had finite expectation. More specifically,

$$x_0^\beta + 1 \le \mathbb{E}[N] \le 2(x_0+1)^\beta + 4 \text{ where } x_0 = \left\lceil \frac{2}{1-(2/3)^{1/\beta}} \right\rceil - 1,$$

which gives that $\mathbb{E}[N] = \exp(\beta \ln(\beta) + \Theta(\beta))$ as $\beta \to \infty$.

Meanwhile, unpublished work of Fawzi [5] while a Master's student of Devroye indicates that they developed independently a different coupling that could be used with dcftp that applies to all β, and in addition they generalized their method to somewhat more general perpetuities. The expected running time in terms of β (called α in their paper) was not analyzed, but the similarity of the coupling to the result in [6] means that it should run in similar time.

Returning to the Dickman distribution, Devroye and Fawzi [3] gave a dctfp algorithm where only 7.96 uniforms were needed on average to generate one Dickman random variable. (Their algorithm requires on average 2.32 steps, each step requires 3 uniforms, and it needs one more uniform to initialize.)

Blanchet and Sigman [1] gave a method that applies when $\ln(W)$ is light-tailed, and A is an unbounded random variable with positive continuous density on $(0, \infty)$. For Vervaat perpetuities $A = 1$ with probability 1, so the method of Blanchet and Sigman does not apply.

1.2. The Markov chain approach

Consider a sequence $U_1, U_2, \ldots$ of iid uniform random variables over $[0, 1]$, with $W_t = U_t^{1/\beta}$ for all t. Form a Markov chain by setting $X_0 = 0$ and $X_{t+1} = W_{t+1}(1 + X_t)$ for all t. After t steps in the Markov chain

$$X_t = W_t + W_t W_{t-1} + \cdots + \prod_{i=1}^{t} W_i,$$

which has the same distribution as

$$Y_t = W_1 + W_1 W_2 + \cdots + \prod_{i=1}^{t} W_i.$$

In other words, the distribution of state t in the Markov chain is the same as the tth partial sum of the perpetuity. How far away is Y_t from Y, the infinite sum that is the perpetuity?

Consider the problem in expectation. From independence of the $\{W_i\}$:

$$\mathbb{E}[Y - Y_t] \geq \sum_{t'=t+1}^{2t} \prod_{i=1}^{t'} \mathbb{E}[W_i] \geq t\mathbb{E}[W_1]^{2t}.$$

Note $\mathbb{E}[W_1] = \mathbb{E}[U_1^{1/\beta}]$ is easily shown to be $(1 - 1/(\beta + 1))$. For $\beta \geq 1$, $(1 - 1/(1 + \beta))^{\beta} \geq e^{-1}$, so $t\mathbb{E}[W_1]^{2t} \geq t \exp(-2t/\beta)$. For $t \in [(1/2)\beta, (1/2)\beta \ln(\beta)]$ this is a decreasing function, so evaluating at $t = (1/2)\beta \ln(\beta)$ gives

$$\mathbb{E}[Y - Y_t] \geq (1/2) \ln(\beta).$$

In other words, for large β, both the partial sums of the perpetuity and the usual Markov chain requires $\Omega(\beta \ln(\beta))$ steps before the state even comes within $\Omega(\ln(\beta))$ of the perpetuity value on average.

This gives evidence (but not proof) that $\Omega(\ln(\beta))$ steps are necessary for any method for this problem that utilizes a Markov chain approach. Of course, a different approach that does not utilize Markov chains might beat this bound. However, currently no method is known to do so. For example, the acceptance/rejection method used by Devroye [2] requires $\Omega(\beta^2)$ uniforms to run. This is shown in Section 3.1.

The goal of this work is to develop a faster method for simulating random variates from the Vervaat perpetuity distribution, particularly for large β.

1.3. The improvement

The original dcftp approach of Fill and Huber [6] only used an upper bound on the state of the Markov chain, and (roughly speaking) requires that steps in the Markov chain must be taken until the state of the process falls below 1.

The expected value of of a Vervaat perpetuity is equal to β. Therefore when $\beta = 1$ much of the density lies below 1. But as β grows, the bulk of the density lies near β, and the area below 1 goes down very fast.

The improvement in the method presented here comes from running both an upper and a lower bound on the Markov chain. Instead of the upper bound needed to fall below 1, it need merely fall below the lower bound of the previous time step. This lower bound starts at 0 but moves towards β on average by a constant amount at each step. Once near β, the upper bound has a significant chance of moving below the lower bound at the previous time step, giving us a sample.

In this way, what was previously an exponential algorithm (because an exponentially small part of the probability resided in the tail below 1) can be turned into a polynomial time algorithm.

Theorem 1. *The algorithm for generating Vervaat perpetuities from Section 2 uses T iid uniform [0,1] random variables, where for all $\beta \geq 1$.*

$$\mathbb{E}[T] \leq (10/3)((\beta + 1)[2\ln(\beta) + \ln(600)] + 1).$$

Section 2 presents the algorithm and proves correctness. Section 3 then proves the running time bound.

2. The method

The method used here is a variant of coupling from the past (cftp). In [6], monotone dominated cftp was employed to draw samples from (1). We begin by presenting the intuition behind Propp and Wilson's monotone cftp [15].

2.1. Monotone cftp

Recall that a Markov chain is a stochastic process such that the next state depends only on the current state, and not on the past history of the process. That means that the next state of the process can be determined from the current state and some independent randomness using a deterministic function called an update function.

Definition 3. For a Markov chain $\{X_i\}$ with state space Ω, $\phi : \Omega \times [0,1] \to \Omega$ is an *update function* if for $U \sim \mathsf{Unif}([0,1])$, $[X_{t+1}|X_t = x_t] \sim \phi(x_t, U)$.

In general update functions can employ much more general randomness than a single uniform, such as multiple uniforms or even an iid sequence of uniforms. For notational simplicity this description supposes that the randomness comes from a single uniform, but everything said here applies to the more general case.

Now suppose that $\preceq$ is a partial order on the state space Ω (so $(a \preceq a)$, $(a \preceq b)$ and $(b \preceq a)$ implies $a = b$, and $(a \preceq b)$ and $(b \preceq c)$ implies $a \preceq c$.)

Definition 4. An update function is *monotonic* for $(\preceq, \Omega)$ if

$$(\forall x, y \in \Omega)(\forall u \in [0,1])(x \preceq y \Rightarrow \phi(x,u) \preceq \phi(y,u)).$$

Let π be a distribution over the state space of the Markov chain. Then say π is stationary if $X_t \sim \pi$ implies $X_{t+1} \sim \pi$ as well. For example, for the Markov chain defined by $X_{t+1} = U_{t+1}^{1/\beta}(1 + X_t)$ where $U_1, U_2, \ldots$ are iid $\mathsf{Unif}([0,1])$, the stationary distribution of the chain is the Vervaat perpetuity. Markov chain Monte Carlo takes advantage of the fact that under limited assumptions, for any fixed x_0, the distribution of X_t approaches π as t goes to infinity.

When applicable, cftp allows the user to directly simulate from π, the stationary distribution of the Markov chain. The intuition is to view the Markov chain as running for times starting from the far past, so for times $\{\ldots, -3, -2, -1, 0\}$. In other words, it has already been running for an infinite amount of time up to time $t = 0$. Having run for an infinite number of steps, the idea is that X_0 comes exactly from the stationary distribution of the Markov chain.

More precisely, fix a time $t < 0$. Suppose that M is the largest state in the state space, and m the smallest, so that $m \preceq x \preceq M$ for all $x \in \Omega$. Let $m_t = m$ and $M_t = M$. For r from $t + 1$ up to 0, let $M_r = \phi(M_{r-1}, U_r)$ and $m_r = \phi(m_{r-1}, U_r)$.

Since $m_t \preceq X_t \preceq M_t$, a simple induction gives that $m_r \preceq X_r \preceq M_r$ for all r up to 0. In particular, if $m_0 = M_0$ then X_0 also equals that common value, and the algorithm terminates with $X_0 \sim \pi$.

If $m_0 \neq M_0$, then recursively call the algorithm to obtain X_t. Then use $U_{t+1}, \ldots, U_0$ to update X_t forward to X_0, and output X_0. Either way, the algorithm will output $X_0 \sim \pi$.

2.2. Dominated cftp

The basic monotone cftp method cannot be used here because the state space is $[0, \infty)$. So no state M is an upper bound on Ω. Kendall and Møller [10, 11] solved this issue by introducing dominating coupling from the past (dcftp), also known as coupling into and from the past.

In their approach, a second update function ϕ_{D} is needed which dominates the original update function in the sense that

$$(\forall x, w \in \Omega)(\forall u \in [0,1])(x \preceq w \Rightarrow \phi(x,u) \preceq \phi_{\text{D}}(w,u)).$$

Call the Markov chain created by update function ϕ_D the *dominating chain*. The chain must have the following properties.

- The dominating chain must have a stationary distribution π_{D}
- It must be possible to draw from π_{D}.
- It must be possible to run the dominating chain D_t backwards from stationarity, that is, to draw D_{t-1} given $D_t \sim \pi$.
- It must be possible to impute the forward U_t values. That is, using the relationship $D_t = \phi_D(D_{t-1}, U_t)$, it must be possible to simulate from $[U_t | D_t, D_{t-1}]$.

With such a dominating chain in place, dcftp runs as follows. Assume $t < 0$ is a parameter given to the program, and that m is a minimum state of the chain, so $(\forall x \in \Omega)(m \preceq x)$.

```
1. Draw $D_0$ from $\pi_D$. Set $t_{\text{old}}$ to 0.
2. For $i$ from $t_{\text{old}} - 1$ down to $t$, draw $D_i$ given $D_{i+1}$.
3. For $i$ from $t + 1$ to $t_{\text{old}}$, draw $U_i$ to be uniform over $[0, 1]$ conditioned on
   $\phi_D(D_{i-1}, U_i) = D_i$.
4. Set $M_t$ to $D_t$, and $m_t$ to $m$.
5. For $i$ from $t + 1$ to 0, let $m_i$ be $\phi(m_{i-1}, U_i)$ and $M_i$ be $\phi(M_{i-1}, U_i)$
6. If $M_0 = m_0$, then output this common value and quit. Else, set $t_{\text{old}}$ to $t$,
   $t$ to $2t$, and return to line 2.
```

Note that by setting t to $2t$ in the last line, the value of t quickly grows to the size needed to have a good chance that $m_0 = M_0$. Setting t to $t + 1$ minimizes the number of random draws that are generated, but if ϕ takes a long time to calculate, this approach can result in a much longer computational time.

2.3. The earlier method

The monotone dominated cftp method of [6] operated as follows. First, the update function used the following fact about uniforms raised to powers.

Lemma 1. *Let $x \geq 0$, and $W = U^{1/\beta}$ where $U \sim \mathsf{Unif}([0, 1])$. Then*

$$[(1 + x)W | (1 + x)W \leq 1] \sim W.$$

Proof. Suppose $\beta = 1$. Then $(1 + x)W \sim \mathsf{Unif}([0, 1 + x])$. Conditioning on $(1 + x)W \leq 1$ is the same as conditioning on $W \leq 1/(1 + x)$. A uniform conditioned on lying in a smaller space is uniform over that smaller space, so $[W | (1 + x)W \leq 1] \sim \mathsf{Unif}([0, 1/(1 + x)])$ which makes $[(1 + x)W | (1 + x)W \leq 1] \sim \mathsf{Unif}([0, 1]) \sim W$.

Now for $\beta \neq 1$. Note $((1 + x)W)^{\beta} \sim U \cdot (1 + x)^{\beta}$. If $(1 + x)W \leq 1$, then $((1 + x)W)^{\beta} \leq 1$. From the $\beta = 1$ case, conditioned on $U(1 + x)^{\beta} \leq 1$, we have $U \cdot (1 + x)^{\beta} \sim \mathsf{Unif}([0, 1])$. So

$$[U \cdot (1 + x)^{\beta} | W(1 + x) \leq 1] \sim U.$$

Raising both sides to the $1/\beta$ power then gives the result. □

This gives rise to the following update function. Draw two uniforms. The first uniform $U(1)$ determines if $U(1)^{1/\beta}(1+x) \leq 1$. If so, then the second uniform $U(2)$ is used to set the next state to $U(2)^{1/\beta}$. Otherwise $U(1)^{1/\beta}(1+x) > 1$, and the next state should be $U(1)^{1/\beta}(1+x)$.

Let $\mathbb{1}(\cdot)$ denote the usual indicator function that evaluates to 1 if the Boolean argument is true, and is 0 otherwise. With this notation, let $S(x,u) = \mathbb{1}(u < 1/(1+x)^{\beta})$, and then the update function becomes

$$\phi(x,u(1),u(2)) = S(x,u(1))u(2)^{1/\beta} + [1 - S(x,u(1))]u(1)^{1/\beta}(1+x). \qquad (3)$$

The key property of this update function, is that if $u(1) < 1/(1+x)^{\beta}$, then the value of $\phi(x,u(1),u(2))$ no longer depends on x! No matter what x is at that point, $\phi(x,u(1),u(2)) = u(2)^{1/\beta}$. So this couples together the process, bringing our bounds on x, which used to form an interval, to the same value. The chance that this coupling occurs is simply the chance that $u(1) < 1/(1+x)^{\beta}$, which is $1/(1+x)^{\beta}$.

Now when β is large, x has to be small before this coupling will occur with reasonable probability. For $U \sim \mathsf{Unif}([0,1])$, $\mathbb{E}[U^{1/\beta}] = \beta/(1+\beta)$. Then equation (2) gives that the expected value of a draw from the perpetuity is β, and $\mathbb{P}(U^{1/\beta}(1+\beta) \leq 1) = (1+\beta)^{-\beta}$. On average, this event takes $(1+\beta)^{\beta}$ steps to occur. This is what leads to the poor running time for large β.

The dominating function for the method is an asymmetric simple random walk on the shifted integers $\{x_0 - 1, x_0, x_0 + 1, \ldots\}$. For $\beta \in (0,\infty)$, let

$$x_0 = \frac{1 + (2/3)^{1/\beta}}{1 - (2/3)^{1/\beta}}$$

and

$$\phi_{\text{D}}(x,u(1),u(2)) = x + \mathbb{1}(u(1) > 2/3) - \mathbb{1}(u(1) \leq 2/3, x \geq x_0)$$

[Note that this is a slight change from the x_0 of [6] that simplifies the algorithm slightly.]

Since the difference between adjacent states of this chain is $\{-1,0,1\}$, this chain is time reversible, which means that from the stationary distribution a simulation forward in time has the same distribution as a simulation backwards in time. Therefore the only question is in evaluating the forward U_t conditioned on D_t and D_{t-1}. This is easy: conditioned on $D_t = D_{t-1}+1$, $U_t \sim \mathsf{Unif}((2/3,1])$. Conditioned on $D_t \leq D_{t-1}$, $U_t \sim \mathsf{Unif}([0,2/3])$.

Lemma 2. *For all $x \in [0,y]$, $y \geq x_0 - 1$, $u(1) \in [0,1]$, and $u(2) \in [0,1]$,*

$$\phi(x,u(1),u(2)) \leq \phi_D(y,u(1),u(2)).$$

Proof. Suppose $u(1) > 2/3$ so $D_t = D_{t-1} + 1$. Then for all x and $u(2)$, it is always true that $\phi(x,u(1),u(2)) \leq 1 + x$, hence the result holds.

Next suppose that $u(1) \leq 2/3$. By the monotonicity of ϕ, $\phi(x,u(1),u(2)) \leq \phi(y,u(1),u(2))$, so it suffices to show that $\phi(y,u(1),u(2)) \leq \phi_{\text{D}}(y,u(1),u(2))$, or equivalently, that

$$u(1)^{1/\beta}(1+y) \leq y - 1.$$

The value of x_0 was chosen so that this inequality is equivalent to $y \geq x_0$, so the result holds.

The last case to consider is when $y = x_0 - 1$. Then $\phi_{\mathrm{D}}(y, u(1), u(2)) = y$. Again, x_0 is large enough that $(2/3)^{1/\beta}(1 + y) \leq y$ when $y = x_0 - 1$, so the result holds for this last case as well. □

2.4. The new method

The new method takes advantage of a simple fact: dominated coupling from the past works even if the update function for the underlying chain is changing from time step to time step. The Markov chain itself is time homogeneous: the distribution of X_t given $X_{t-1} = x$ is unchanging with t. However, the update function used to move the chain can be changing from step to step as long as each update function is still an update function for the original chain, and the dominating chain is still dominating at each step. That is, it is important to have a family of update functions ϕ_t such that:

$$(\forall t)(\forall x \in \Omega)(U \sim \mathsf{Unif}([0,1]) \rightarrow \phi_t(x, U) \sim [X_{t+1} | X_t = x])$$

and

$$(\forall t)(\forall x \preceq w)(\forall u \in [0,1])(\phi_t(x, u) \preceq \phi_{\mathrm{D}}(w, u))$$

To take advantage of this flexibility, we need to generalize Lemma 1.

Lemma 3. *For any $x > 0$ and $a \geq 1$,*

$$[W(1 + x) | W(1 + x) \leq a] \sim aU^{1/\beta}$$

Proof. Let $U_1 \sim \mathsf{Unif}([0,1])$ and $W = U_1^{1/\beta}$. Then $W(1 + x) \leq a \Rightarrow U_1 \in [0, (a/(1 + x))^\beta]$. Let $U_2 \sim \mathsf{Unif}([0, (a/(1 + x))^\beta])$. Then $[W(1 + x)|W(1 + x) \leq a] \sim U_2^{1/\beta}(1 + x)$. For $U_3 \sim \mathsf{Unif}([0,1])$, $U_3(a/(1 + x))^\beta \sim U_2$. So

$$[W(1 + x)|W(1 + x) \leq a] \sim (U_3(1/(1 + x))^\beta)^{1/\beta}(1 + x) = aU_3,$$

which completes the proof. □

Now suppose it is known that $x_t \leq M_t$. Then the update function at time t depends on M_t. First, update M_t as before to get

$$M_{t+1} = u^{1/\beta}(1 + M_t). \tag{4}$$

Note that Lemma 3 gives us that

$$[M_{t+1} | M_{t+1} \in [0, 1 + x_t]] \sim [x_{t+1} | x_t].$$

So, in the case when $M_{t+1} \leq (1 + x_t)$, we may define our new update function ϕ_t such that $\phi_t(x_t, u) = M_{t+1}$.

If $M_{t+1} > 1 + x_t$, then the uniform u must be between $[(1 + x_t)/(1 + M_t)]^\beta$ and 1. So shifting this uniform downward and rescaling so that it lies in $[0, 1]$ gives a uniform that can be used to update x_t. Therefore, let

$$u' = \left[\frac{u - [(1 + x_t)/(1 + M_t)]^\beta}{1 - [(1 + x_t)/(1 + M_t)]^\beta} \right]. \tag{5}$$

Then

$$[u' | M_{t+1} > 1 + x_t] \sim \mathsf{Unif}([0, 1]).$$

Taken together, this gives the following update function. For M_{t+1} and u' as in (4) and (5), $\phi_t(x, u) = \phi(x, u, M_t)$ defined as

$$\phi(x, u, M_t) = M_{t+1}\mathbb{1}(M_{t+1} \le 1 + x_t) + u'^{1/\beta}(1 + x_t)\mathbb{1}(M_{t+1} > 1 + x_t) \tag{6}$$

2.5. The complete algorithm in pseudocode

Unless specifically mentioned, all random variable draws are taken to be independent.

1. Inputs are β (the parameter of the Vervaat family), ℓ (the initial number of steps to run), and an optional input D_0 (the value of the dominating chain at time 0).
2. Initialize by setting $x_0 \leftarrow (1 + (2/3)^{1/\beta})/(1 - (2/3)^{1/\beta})$.
3. If D_0 is given as an input to the algorithm, use it, otherwise, let $D_0 \leftarrow x_0 - 1 + G$, where $G \sim \mathsf{Geo}(1/2)$.
4. Generate $D_{-1}, D_{-2}, \ldots, D_{-\ell}$ using a reversible asymmetric simple random walk with partially reflecting boundary at $x_0 - 1$. That is, for t from -1 down to $-\ell$, draw $A \sim \mathsf{Unif}([0, 1])$, then let

$$D_{t-1} \leftarrow D_t + \mathbb{1}(A > 2/3) - \mathbb{1}(A \le 2/3, D_t \ge x_0).$$

5. Set $m_{-\ell}$ equal to 0 and $M_{-\ell}$ equal to $D_{-\ell}$. For t from $-\ell + 1$ up to 0, draw U_t as $\mathsf{Unif}([0, 2/3])$ if $D_t \le D_{t-1}$, or as $\mathsf{Unif}((2/3, 1])$ if $D_t > D_{t-1}$.
6. For t from $-\ell + 1$ to 0, let

$$m_t \leftarrow \phi(m_{t-1}, U_t, M_{t-1}), M_t \leftarrow \phi(M_{t-1}, U_t, M_{t-1}).$$

7. If $m_0 = M_0$, output this common value as the result and quit. Otherwise, perform the following steps. First, call the algorithm recursively, with input for the dominating chain of $D_{-\ell}$, and time steps 2ℓ (β remains the same). Call the outcome of the recursive call Y. Next, set $m_{-\ell} \leftarrow Y$ and $M_{-\ell} \leftarrow D_{-\ell}$. Then for t from $-\ell + 1$ to 0, let $m_t \leftarrow \phi(m_{t-1}, U_t, M_{t-1})$, $M_t \leftarrow \phi(M_{t-1}, U_t, M_{t-1})$. Finally, output m_0 and quit.

This procedure is implemented in R in the appendix.

The new algorithm differs in two key ways from dominated cftp of the previous section. First (and most importantly), it maintains both an upper and a lower bound on the process, thereby speeding coalescence. Second, since the algorithm is being called recursively, the values of the lower and upper process are reset when the recursive call ends.

So for example, suppose initially $t = 10$. Then the process is run over 10 steps, from time $t = -10$ up to $t = 0$. In the recursive call, twice as many steps are used, so in this example 20 steps. This can be viewed as generating the value of the process from time $t = -30$ up to $t = -10$. If coalescence occurs at or before time $t = -10$, when the process reaches time -10, m_{-10} is still reset to 0, and M_{-10} is reset to the value of the dominating process before the state is run forward up to time 0. This ensures that every step is updated according to the same update function, using the same random choices.

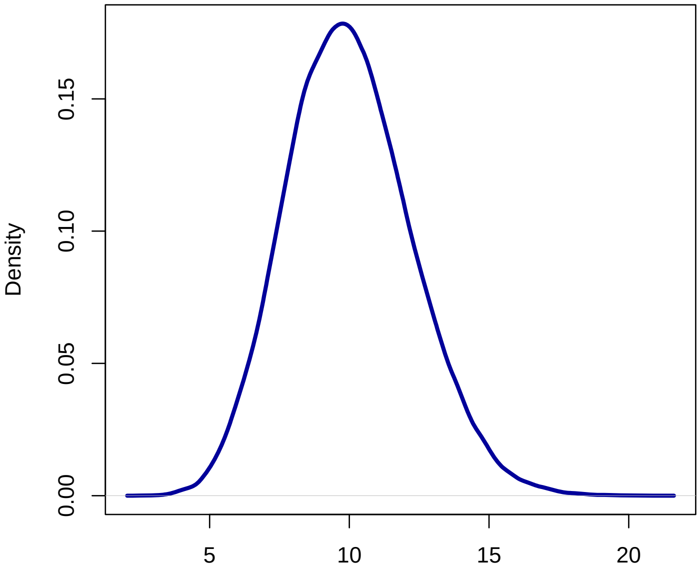

Figure 1: Density of Vervaat perpetuity with $\beta = 10$ estimated from 10,000 samples. A single sample from this β would have required at least 10^{16} uniforms (on average) using the older method of Fill & Huber [6, Theorem 1]. The new method of this paper used on average about 217 uniforms per sample, and 10,000 samples were generated in 23 seconds using R on a Windows 10 machine with an Intel Core i7-6600U CPU at 2.60GHz.

3. The run time

Now consider the average number of steps taken by the algorithm. This is proportional to the largest value of ℓ input to the algorithm, and so to understand the running time it is necessary to under how large ℓ grows on average.

Lemma 4. *For $D_0 = x_0 - 1 + G$, and $\ell \geq \ln(4\delta^{-2}\beta(x_0 + 1))(1 + \beta) + 1$ for $\delta > 0$, the chance that the algorithm returns Y at the first step is at least $1 - \delta$.*

Proof. The expected value of D_0 is $x_0 + 1$. Let $m \leq M$ and consider $m' = \phi(m, U_t, M)$, $M' = \phi(M, U_t, M)$. Then

$$\mathbb{E}[M' - m'|m, M] = \mathbb{E}[U^{1/\beta}(1 + M)|M] - \mathbb{E}[U^{1/\beta}(1 + m)|m] = \frac{\beta}{\beta + 1}(M - m).$$

At the beginning of line 6, $M - m = D_{-\ell} - 0$. Now let M_{-1} and m_{-1} be the values of M and m at the end of $\ell - 1$ steps in line 6. Then a simple induction gives

$$\mathbb{E}[M_{-1} - m_{-1}|D_{-\ell}] = \left(\frac{\beta}{\beta + 1}\right)^{\ell-1} D_{-\ell}.$$

Taking the expected value of both sides and $\beta/(\beta + 1) \leq \exp(-1/(\beta + 1))$ gives

$$\mathbb{E}[M_{-1} - m_{-1}] \leq (x_0 + 1)\exp(-(\ell - 1)/(\beta + 1)).$$

Then

$$(\forall a > 0)(\mathbb{P}(M_{-1} - m_{-1} \geq a) \leq a^{-1}(x_0 + 1)\exp[-(\ell - 1)/(\beta + 1)])$$

by Markov's inequality.

When $U_0 \leq ((1+m)/(1+M))^\beta$, then $m_0 = M_0$ at the final step, so the next goal is to show that $((1+m)/(1+M))^\beta$ is close to 1. Suppose $((1+m)/(1+M)) \geq 1 - (1/2)\delta/\beta$. It is easy to show that for $\beta \geq 1$,

$$\left(1 - \frac{\delta}{2\beta}\right)^\beta \geq 1 - \frac{\delta}{2}.$$

Since $(1 + m)/(1 + M) = 1 - (M - m)/(1 + m)$ and $m \geq 0$, this means

$$\mathbb{P}(m_0 = M_0|M - m \leq (1/2)\delta/\beta) \geq 1 - \delta/2. \tag{7}$$

Suppose $\ell \geq \ln(4\delta^{-2}\beta(x_0 + 1))(\beta + 1) + 1$. Then the chance that $M - m > (1/2)\delta/\beta$ is at most $2\beta\delta^{-1}(x_0 + 1)\exp(-\ln(4\delta^{-2}\beta(x_0 + 1))) = \delta/2$, which together with (7) completes the proof. □

For $\beta \geq 1$, it holds that $\ln(x_0 + 1) \leq \ln(6\beta)$. Suppose that

$$\ell = (\beta + 1)[2\ln(\delta^{-1}) + \ln(4) + \ln(\beta) + \ln(6\beta)] + 1.$$

Then it holds that

$$2\ell \geq (\beta + 1)[2\ln(\delta^{-2}) + \ln(24) + 2\ln(\beta)] + 1.$$

That is to say, if ℓ gives a chance of failure to recurse of δ, then 2ℓ steps gives a chance of failure to recurse of at most δ^2. Hence if the initial ℓ satisfies the inequality with $\delta = 1/5$, then 2ℓ satisfies the inequality with $\delta = (1/5)^2 = 1/25$, and so forth.

Lemma 5. *Let T be the sum of all the values of ℓ in the inputs to all the calls to the algorithm. If $\beta \geq 1$, then*

$$\mathbb{E}[T] \leq (5/3)((\beta+1)[2\ln(\beta)+\ln(600)]+1).$$

Proof. Let $\ell_i = 2^i$, where i is the depth of the recursion. Then if R is the greatest level of recursion called (and recursion level 0 refers to the initial call to the algorithm), then $T = \sum_{i=0}^{R} 2^i = \sum_{i=0}^{\infty} 2^i \cdot \mathbb{1}(i \leq R)$, so

$$\mathbb{E}[T] = \sum_{i=0}^{\infty} 2^i \cdot \mathbb{P}(i \leq R).$$

Let $n = (\beta+1)[2\ln(5)+\ln(24)+2\ln(\beta)]+1$. Then for $i < \log_2(n)$, $\mathbb{P}(i \leq R) \leq 1$. The sum of 2^i for these terms is at most n.

From the previous discussion $\mathbb{P}(\lceil \log_2(n) \rceil \leq R) = 1/5$, $\mathbb{P}(\lceil \log_2(n) \rceil + 1 \leq R) \leq 1/25$, and so on. The sum of these $2^i\mathbb{P}(i \leq R)$ terms is $2n/5+4n/25+\cdots = (2/3)n$. Therefore $\mathbb{E}[T] \leq (5/3)n$, which completes the proof of the bound of the mean. □

Note that each step of the chain uses two uniforms, one for the backward dominating chain, and one for the forward chain. Therefore the bound on the number of uniforms in Theorem 1 is twice the bound on the number of steps from the previous lemma.

Higher moments of T can be bounded similarly.

Lemma 6. *For T as before,*

$$\mathbb{E}[T^2] \leq (76/3)((\beta+1)[2\ln(\beta)+\ln(600)]+1)^2.$$

Proof. For n and R as in the previous proof, and $T = \sum_{i=0}^{\infty} 2^i \mathbb{1}(i \leq R)$, then

$$T^2 = \sum_{i=0}^{\infty}\sum_{j=0}^{i} 2 \cdot 2^i 2^j \mathbb{1}(i \leq R)\mathbb{1}(j \leq R) \leq \sum_{i=0}^{\infty} 2^{2i+2}\mathbb{1}(i \leq R).$$

Now the sum of the expectations of the individual terms is $4(4/3)n^2$ for $i < \log_2(n)$, and $4[(4/5)n^2 + (16/25)n^2 + \cdots] = 20n^2$ for the rest of the terms, giving the result. □

3.1. Run time measured by uniforms used

In order to test how this new method compares in practice to methods in [2, 6, 3] we compared how many uniforms would be used in practice.

The results are presented in Figure 2.

When β is small, there is not much difference between the new method and the results of [6]. This is because the chain is very likely to become small. The method of Devroye and Fawzi [3] for $\beta = 1$ uses a better initial upper bound on the perpetuity, which leads to slightly fewer steps in the Markov chain needed to obtain a sample.

β	1	2	3	4	5	10
Fill & Huber (2010)	10.8	27.3	101.6	484	3490	-
this paper	10.1	27.1	47.3	68.7	91.3	217
Devroye & Fawzi (2010)	7.94					
Devroye (2001)	5.65	12.5	22.6	36.29	53.19	291.9

Figure 2: The results from this paper and Fill & Huber (2010) report the sample average of the number of uniform $[0, 1]$ random variables needed to obtain a single draw from the perpetuity. The average used 10,000 independent runs for Fill & Huber (2010) and 100,000 runs for the other three approaches. The Devroye & Fawzi (2010) method only applies to $\beta \leq 1$, and so only the number of uniforms needed for $\beta = 1$ is given. The Devroye (2001) method only includes the average number of uniform [0,1] random variables used, it does not include the extra computation needed to approximate the sine integral function or determine acceptance or rejection.

Finally, we come to the first method of Devroye [2], which at first glance appears to be the best method. However, this algorithm is very different from the others, which use dominated coupling from the past. In these methods, the computational effort required per uniform value is very small.

The method of [2] utilizes an acceptance rejection method to draw from the density of the perpetuity. However, it is not possible to directly calculate the target density exactly. Instead, a sequence of approximations is used which gets provably close to the true value. This requires far most computation than a Markov chain step, and much of the paper is devoted to showing that this extra computation needed can be accomplished in time that has finite expectation for large β. Moreover, the number of uniform $[0, 1]$ random variables used by the method grows quadratically in β.

Lemma 7. *When $\beta \geq 1$, the expected number of uniforms used by the 2001 method of Devroye [2] is at least*

$$[3/(e-1)]\beta(\beta+1).$$

Proof. The acceptance rejection method bounds the target density by

$$h(x) = \min\{\beta(\beta+1)/x^2, \beta x^{\beta-1}\}.$$

As is well known (see for instance [7], the expected number of draws required by an acceptance rejection method for a normalized density of a nonnegative random variable upper bounded by an unnormalized envelope function is

$$\mathbb{E}[T] = \int_0^\infty h(x)\ dx$$

When $x \geq (\beta+1)^{1/(\beta+1)}$, then $h(x) = \beta(\beta+1)/x^2$. The integral of this last function from $(\beta+1)^{1/(\beta+1)}$ to infinity is

$$\beta(\beta+1)\frac{1}{(\beta+1)^{1/(\beta+1)}}.$$

The last factor is at least $1/(e-1)$ for $\beta \geq 1$.

Each acceptance rejection trial uses 3 uniforms ([2, p. 106]), which completes the proof. □

4. Acknowledgments

This work was supported by NSF DMS-1418495.

Appendix A. Implementation

The following code implements the Vervaat perpetuity algorithm given in pseudocode in Section 2.5 for R version 3.0.2 (2013-09-25).

```
vervaat.time <- function(beta = 1,steps = 1,d = -1) {
# Written by Alex Cloud and Mark Huber
# 21 March 2017

  # Line 2
  x0 <- (1+(2/3)^(1/beta))/(1-(2/3)^(1/beta))
  # Line 3
  if (d == -1) d <- x0 - 1 + rgeom(1,prob=1/2)
  # Line 4
  d <- c(rep(0,steps),d); a <- runif(steps)
  for (t in steps:1) d[t] <- d[t+1]+(a[t]>2/3)-(a[t]<=2/3)*(d[t+1]>=x0)
  # Line 5
  m <- 0;M <- d[1]; u1 <- rep(0,steps); u2 <- runif(steps)
  for (t in 2:(steps+1)) {
    up <- d[t]>d[t-1]; u1[t - 1] <- runif(1,min=2/3*up,max=2/3+1/3*up)
  }
  # Line 6
  for (t in 2:(steps+1)) {
    Mold <- M
    M <- u1[t-1]^(1/beta)*(1+M)
    if (M < (m+1))
      m <- M
    else {
      r <- ((m+1)/(Mold+1))^beta
      m <- ((u1[t-1] - r)/(1-r))^(1/beta)*(1+m)
    }
  }
  # Line 7
  value <- c(0,0)
  if (m == M) return(c(m,steps)) else {
    value <- vervaat.time(beta,2*steps,d[1]); m <- value[1]; M <- d[1]
    for (t in 2:(steps+1)) {
      Mold <- M
      M <- u1[t-1]^(1/beta)*(1+M)
      if (M < (m + 1))
        m <- M
      else {
        r <- ((m+1)/(Mold+1))^beta
        m <- ((u1[t-1] - r)/(1-r))^(1/beta)*(1+m)
      }
    }
    return(c(m,value[2]+steps))
  }
}
```

[1] J. H. Blanchet and K. Sigman. On exact sampling of stochastic perpetuities. *J. Appl. Prob. Spec.*, 48A:165–182, 2011.

[2] L. Devroye. Simulating perpetuities. *Methodol. Comput. Appl. Probab.*, 3(1):97–115, 2001.

[3] L. Devroye and O. Fawzi. Simulating the Dickman distribution. *Statist. Probab. Lett.*, 80(3–4):242–247, 2010.

[4] K. Dickman. On the frequency of numbers containing prime factors of certain relative magnitude. *Arkiv fr̈ Mathematik, Astronomi och Fysik*, 22:1–14, 1930.

[5] O. Fawzi. Efficient sampling of perpetuities using coupling from the past, 2007. Retrieved from `http://perso.ens-lyon.fr/omar.fawzi/docs/rapportM1.pdf` on Feb 5, 2017.

[6] J. A. Fill and M. L. Huber. Perfect simulation of Vervaat perpetuities. *Electron. J. Probab.*, 15:96–109, 2010.

[7] G. S. Fishman. Choosing sample path length and number of sample paths when starting in the steady state. *Oper. Res. Letters*, 16:209–219, 1994.

[8] C.R. Hoare. Find (algorithm 65). *Comm. ACM*, 4:321–322, 1961.

[9] H.-K. Hwang and T.-H. Tsai. Quickselect and dickman function. *Combin. Probab. Comput.*, 11(4):353–371, 2002.

[10] W. S. Kendall. Perfect simulation for the area-interaction point process. In *Proceedings of the Sympos. on Probability Towards the Year 2000*, 1995.

[11] W. S. Kendall and J. Møller. Perfect simulation using dominating processes on ordered spaces, with application to locally stable point processes. *Adv. Appl. Prob.*, 32:844–865, 2000.

[12] F. Lassner. *Sommes de produits de variable aléatories indṕendantes*. PhD thesis, Université de Paris VL, 1974.

[13] F. Lessner. Sur certains types de mécanismes additifs en economie stochastique. *C.R. Acad. Sci. Paris*, A279:33–36, 1974.

[14] H. M. Mahmoud, R. Modarres, and R.T. Smythe. Analysis of quickselect: An algorithm for order statistics. *RAIRO Inform. Théor. Appl*, 29:255–276, 1995.

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

[16] V.R.R. Uppuluri, P.I. Feder, and L.R. Shenton. Stochastic difference equations occurring in one-compartment models. *Math. Biosciences*, 1:143–171, 1967.

[17] W. Vervaat. On a stochastic difference equation and a representation of non-negative infinite divisible random variables. *Adv. in Appl. Probab.*, 11(4):750–783, 1979.