Fast, efficient on-line simulation of self-similar processes; Corrected version

Owen Dafydd Jones
University of Melbourne

2004

Abstract

We describe a class of self-similar processes that can be used to fit self-similar data, and give a fast, efficient on-line algorithm for simulating them.

1 Introduction

Self-similar processes are of interest as models for internet packet arrival data, high-frequency financial data, ECG and EEG traces, and various hydrological and meteorological time series. Simulation of self-similar processes has proven problematic, because they exhibit a slowly decaying correlation structure (long-range dependence), which means that the individual elements of any sequence of observations $X(1), \ldots, X(n)$ are strongly correlated. In practice to date this generally means that it is not possible to simulate $X(n)$ without simultaneously simulating $X(1), \ldots, X(n-1)$, and this necessarily results in an algorithm that requires $O(n)$ storage to generate n steps of X. More importantly, this means that if you have already generated n steps, then it is not possible to generate step $n+1$ directly, instead it is necessary to generate all $n+1$ steps from scratch.[1, 2, 3]

One model which avoids these problems is the $M/G/\infty$ queue.[4] Unfortunately this model is not flexible enough to be useful in practice. Here we present a new class of self-similar models called EBP processes (for Embedded Branching Process), which are flexible, readily fitted to data, and easily simulated. Features of the simulation algorithm are

(i) Scaleable: $O(n \log n)$ time to generate n steps.

(ii) Efficient: $O(\log n)$ storage required to generate n steps.

(iii) On line: can generate a new step on demand.

The class of processes is described in Section 2, and the simulation algorithm given in Section 3. A `matlab` implementation of the algorithm can be found on the authors web page at `www.maths.soton.ac.uk/staff/ODJones/`. Some examples are given in Section 3.

2 EBP processes

Suppose that $X : \mathbb{R}_+ \to \mathbb{R}$ is a continuous process. X is said to be self-similar if for some H and all $a > 0$

$$X(t) = a^{-H} X(at) \text{ in distribution.}$$

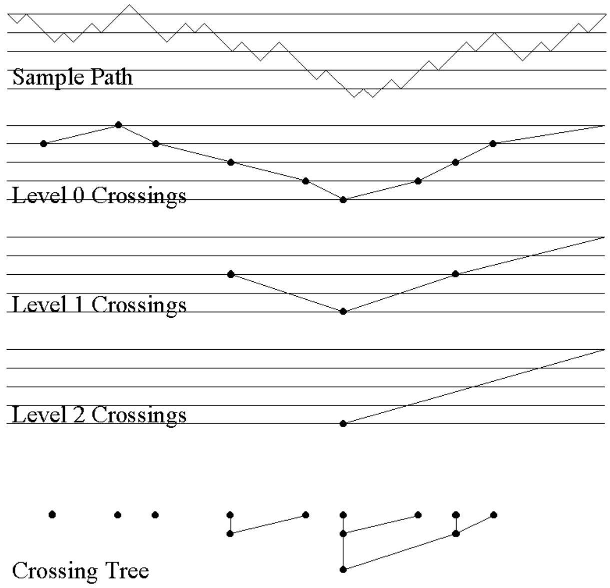

Figure 1: An example of a crossing tree.

H is called the *Hurst index*.

We construct the diadic crossing tree for X. W.l.o.g. suppose $X(0) = 0$. For any $n \in \mathbb{Z}$, let $T_0^n = 0$ and $T_{k+1}^n = \inf\{t > T_k^n : X(t) \in 2^n\mathbb{Z}, X(t) \neq X(T_k^n)\}$ be the hitting times of the size 2^n crossings of the process. The path from $X(0)$ to $X(T_1^n)$ is generally not a true crossing, so we discard it. That is, the k-th size 2^n crossing is from $X(T_k^n)$ to $X(T_{k+1}^n)$.

There is a natural tree structure to the crossings, as each crossing of size 2^n can be decomposed into a sequence of crossings of size 2^{n-1}. The nodes of the crossing tree are crossings, and the offspring of any given crossing are the corresponding set of subcrossings at the level below. Let Z_k^n be the number of subcrossings of size 2^{n-1} that make up the k-th crossing of size 2^n. A crossing tree is illustrated in Fig. 1. Note that the crossing tree is well defined for any continuous process, not just self similar processes.

If X is self-similar with stationary increments, then it can be shown that the Z_k^n form a stationary sequence. Conversely, we will call any continuous process X an *Embedded-Branching-Process* (EBP) process if the Z_k^n are independent and identically distributed. In this case the tree descending from any fixed crossing is a realisation of a Galton-Watson branching process.

From now on let $X(t)$ be an EBP process, and let $p(x) = \mathrm{P}(Z_k^n = x)$ be the *offspring distribution*. Clearly p satisfies $\{x : p(x) > 0\} \subset \{2, 4, 6, \ldots\}$. If in addition we have

$$p(2) < 1 \text{ and } \sum_{x=1}^{\infty} x \log(x)\, p(x) < \infty,$$

then we say that p is *regular*.

Theorem 1 *For any regular discrete distribution p there exists a continuous EBP process X for which p is the offspring distribution. Let $\mu = \sum_{x=1}^{\infty} x\, p(x)$ and $H = \log 2/ \log \mu$,* *then for all* $a = \mu^n$, $n \in \mathbb{Z}$

$$X(t) = a^{-H} X(at) \text{ in distribution.} \tag{1}$$

The proof is deferred to Section 4.

2.1 Fitting to data

Fitting an EBP process requires an estimate of the subcrossing distribution p. This is readily achieved by computing the crossing tree, and then using the observed subcrossing numbers Z_k^n to estimate p. One can easily test the assumption that the Z_k^n are i.i.d., and in practice this is seen to be a reasonable assumption to make for self-similar data. Thus EBP processes form an extremely flexible class of models for self-similar processes.

An application of the crossing tree to the estimation of the Hurst index H of a self-similar process is given by Jones & Shen.[5]

2.2 Markov representation

Let X^m be the random walk on $2^m\mathbb{Z}$ obtained by observing only 2^m crossings of X. That is $X^m(k) = X(T_k^m)$ for $k = 1, 2, \ldots$. In this section we give an infinite dimensional representation of X^m which is Markov.

Let C_k^n be the k-th crossing of size 2^n. By a crossing we mean a section of the sample path, plus some extra information such as the time and place the crossing starts. We adopt the convention that a crossing includes its initial point but not its final point. For $n \geq m$ let $\kappa(m, n, k)$ be such that $X^m(k) \in C^n_{\kappa(m,n,k)}$, for $k = 1, 2, \ldots$. We have that C_k^n has Z_k^n subcrossings, and define S_k^n to be such that C_k^n is subcrossing number S_k^n of $C^{n+1}_{\kappa(n,n+1,k)}$. Clearly $1 \leq S_k^n \leq Z^{n+1}_{\kappa(n,n+1,k)}$.

A crossing has one of 6 types depending upon its direction (up or down) and where it starts from. Suppose that we have a crossing of size 2^n, and that the parent crossing starts from $k2^{n+1}$. The 6 types are then $0^+, 0^-, 1^+, 1^-, -1^+, -1^-$, where a type i^+ crossing is from $k2^{n+1} + i2^n$ to $k2^{n+1} + (i+1)2^n$, and a type i^- crossing is from $k2^{n+1} + i2^n$ to $k2^{n+1} + (i-1)2^n$. Let α_k^n be the type of crossing C_k^n.

We define the *crossing state* of X^m at time k to be $\mathcal{X}^m(k) = \{\mathcal{X}^{m,n}(k)\}_{n \geq m}$ where

$$\mathcal{X}^{m,n}(k) = (\kappa(m, n, k), S^n_{\kappa(m,n,k)}, Z^{n+1}_{\kappa(m,n+1,k)}, \alpha^n_{\kappa(m,n,k)}).$$

We will occasionally write $S^{m,n}(k)$, $Z^{m,n+1}(k)$ and $\alpha^{m,n}(k)$ for $S^n_{\kappa(m,n,k)}$, $Z^{n+1}_{\kappa(m,n+1,k)}$ and $\alpha^n_{\kappa(m,n,k)}$. If $S^{m,n}(k) = Z^{m,n+1}(k)$ then $X^m(k)$ is at the end of a level $n+1$ crossing.

Theorem 2 $\mathcal{X}^m$ *is a Markov chain.*

Proof We describe how to generate $\mathcal{X}^m(k+1)$ from $\mathcal{X}^m(k)$ using the recursive procedure `Increment` acting on the $\mathcal{X}^{m,n}(k)$. `Increment` is applied to $\mathcal{X}^{m,n}(k)$ when step $k+1$ of X^m takes it into a new crossing at level n.

> `Procedure Increment` $\mathcal{X}^{m,n}(k)$
>
> (Assume that $X^m(k)$ is at the end of a level n crossing)
>
> (This is always the case for $n = m$)
>
> $\kappa(m, n, k+1) = \kappa(m, n, k) + 1$

```
If $S^n_{\kappa(m,n,k)} = Z^{n+1}_{\kappa(m,n+1,k)}$ Then ($X^m(k)$ at end of level $n+1$ crossing)
    Increment $\mathcal{X}^{m,n+1}(k)$
    $S^n_{\kappa(m,n,k+1)} = 1$
    Generate $Z^{n+1}_{\kappa(m,n+1,k+1)}$ using distribution $p$
Else ($X^m(k)$ not at end of level $n+1$ crossing)
    $\mathcal{X}^{m,q}(k+1) = \mathcal{X}^{m,q}(k)$ for all $q \geq n+1$        (†)
    $S^n_{\kappa(m,n,k+1)} = S^n_{\kappa(m,n,k)} + 1$
    $Z^{n+1}_{\kappa(m,n+1,k+1)} = Z^{n+1}_{\kappa(m,n+1,k)}$
End If
(Now determine the type of the new level $n$ crossing)
If $S^n_{\kappa(m,n,k+1)} = Z^{n+1}_{\kappa(m,n+1,k+1)}$ Then        (‡)
    If $\alpha^{n+1}_{\kappa(m,n,k+1)} = i^+$ Then
        $\alpha^n_{\kappa(m,n,k+1)} = 1^+$
    Else
        $\alpha^n_{\kappa(m,n,k+1)} = -1^-$
    End If
Else If $S^n_{\kappa(m,n,k+1)} = Z^{n+1}_{\kappa(m,n+1,k+1)} - 1$ Then
    If $\alpha^{n+1}_{\kappa(m,n,k+1)} = i^+$ Then
        $\alpha^n_{\kappa(m,n,k+1)} = 0^+$
    Else
        $\alpha^n_{\kappa(m,n,k+1)} = 0^-$
    End If
Else If $S^n_{\kappa(m,n,k+1)}$ is odd Then
    $\alpha^n_{\kappa(m,n,k+1)} = 0^+$ or $0^-$ with equal probability
Else
    If $\alpha^n_{\kappa(m,n,k)} = 0^+$ Then $\alpha^n_{\kappa(m,n,k+1)} = 1^-$ End If
    If $\alpha^n_{\kappa(m,n,k)} = 0^-$ Then $\alpha^n_{\kappa(m,n,k+1)} = -1^+$ End If
End If
End Procedure
```

To update $\mathcal{X}^m(k)$ to $\mathcal{X}^m(k+1)$ we apply procedure `Increment` to $\mathcal{X}^{m,m}(k)$. `Increment` is recursively applied to all $\mathcal{X}^{m,n}(k)$ such that $S^q_{\kappa(m,q,k)} = Z^{q+1}_{\kappa(m,q+1,k)}$ for all $m \leq q < n$. For all n larger than this we get $\mathcal{X}^{m,n}(k+1) = \mathcal{X}^{m,n}(k)$.

Procedure `Increment` will always terminate after a finite number of recursive calls, provided we do not have $S^q_{\kappa(m,q,k)} = Z^{q+1}_{\kappa(m,q+1,k)}$ for all $q \geq m$. However, if this is the case then for all $n \geq m$ we put $S^n_{\kappa(m,n,k+1)} = 1$, generate $Z^{n+1}_{\kappa(m,n+1,k+1)}$ according to the distribution p (independently of each other), and then generate types consistently.

Since `Increment` only requires $\mathcal{X}^m(k)$ to generate $\mathcal{X}^{m+1}(k)$, $\mathcal{X}^m$ is a Markov process. □

Clearly, given $\mathcal{X}^{m,m}(k)$ we know $X^m(k+1) - X^m(k)$. So if we can simulate $\mathcal{X}^{m,m}$ then we can simulate X^m.

Note that $\mathcal{X}^m$ has a countable state space. Clearly it is transient if we include $\kappa(m,n,k)$ in the crossing state, as $\kappa(m,n,k)$ is a non-decreasing function of k. However, the chain is still transient if we remove the $\kappa(m,n,k)$. To see this, suppose we write $\{s_m, s_{m+1}, s_{m+2}, \ldots\} < \{t_m, t_{m+1}, t_{m+2}, \ldots\}$ if $s_n = t_n$ for all $n > n_0$ and $s_{n_0} < t_{n_0}$, for some $n_0 < \infty$. This defines a partial ordering on $\mathbb{Z}_+^{\mathbb{Z}_+}$. We have $\{S^m_{\kappa(m,m,k)}, S^{m+1}_{\kappa(m,m+1,k)}, \ldots\} <$

$\{S^m_{\kappa(m,m,k+1)}, S^{m+1}_{\kappa(m,m+1,k+1)}, \ldots\}$ for all k. Thus, not only is $\mathcal{X}^m$ transient, every state is visited at most once.

3 Simulation algorithm

The procedure `Increment` used to prove Theorem 2 is used as the basis of an algorithm to generate X^m. As noted previously, we can generate X^m from $\mathcal{X}^{m,m}$. The crux of our algorithm is the observation that we do not need all of $\mathcal{X}^m(k)$ to generate $\mathcal{X}^{m,m}(k+1)$. Instead we can use a truncated version $\{\mathcal{X}^{m,m}(k), \ldots, \mathcal{X}^{m,n}(k)\}$, where $n = O(\log k)$. This is because we only need to know $\mathcal{X}^{m,n+1}(k)$ when there is a new level $n+1$ crossing, and the first level n crossing occurs after $O(e^n)$ level m steps.

One way of achieving this is to put $S^{m,n}(1) = S^n_1 = 1$ for all n. In this case, if the first level n crossing ends at step k then $\mathcal{X}^{m,n}(k) = (1, 1, Z^{n+1}_1, \alpha^n_1)$ where the Z^{n+1}_1 are i.i.d. with distribution p. If $\alpha^{m,n-1}(k) = 1^+$ then $\alpha^{m,n}(k) = \alpha^n_1 = 0^+$. If $\alpha^{m,n-1}(k) = -1^-$ then $\alpha^{m,n}(k) = \alpha^n_1 = 0^-$.

While this provides a method of generating $\mathcal{X}^{m,n}(k)$ when it is first required, in practice this approach is undesirable. The correlation structure of X^m is determined by the branching structure of the crossing tree. Restricting $\mathcal{X}^m(1)$ as above effectively means we are conditioning X^m in some manner. As $\mathcal{X}^m$ is transient, it does not have an equilibrium distribution, so we can not choose $\mathcal{X}^m(1)$ in equilibrium. None-the-less, we can still choose $\mathcal{X}^{m,n}(k)$ in a random fashion. The question we need to ask is: "for fixed n, if we observe an EBP process at a 'random' point in time k, what is the distribution of $\mathcal{X}^{m,n}(k)$?"

Suppose that we have a sequence of i.i.d. non-negative random variables arranged into families, $X_{1,1}, X_{1,2}, \ldots, X_{1,N(1)}, X_{2,1}, \ldots, X_{k,N(k)}, \ldots$, where $\mathrm{P}(X_{i,j} \le x) = F(x)$ and $\mathrm{P}(N(i) = n) = p(n)$. Partition $[0, \infty)$ into adjacent intervals $[a, b)$ with lengths $X_{i,j}$, ordered as above. Then choose a point x 'uniformly' in $[0, \infty)$ and consider the size of the interval and family that contain x. Here when we say x chosen uniformly in $[0, \infty)$ we mean in the limit as $T \to \infty$ for x chosen uniformly in $[0, T)$.

We think of the $X_{i,j}$ as level n crossing times and $\sum_{j=1}^{N(i)} X_{i,j}$ as level $n+1$ crossing times.

Lemma 3 *Let $\mathcal{P}$ be the partition above, let X^* and N^* be the interval length and family size of a 'uniformly' chosen point, and let J^* be the position of the chosen interval within its family. If $\mu = \sum_x x\, p(x)$ and $m = \mathrm{E}X_{i,j}$ are finite then with probability 1, for $1 \le l \le n$*

$$\mathrm{P}(N^* = n,\, J^* = l,\, x \le X^* < x + dx \,|\, \mathcal{P}) = \frac{np(n)}{\mu} \frac{1}{n} \frac{x}{m} dF(x).$$

The proof is given in Section 4

At important consequence of this result is that N^* and J^* are independent of X^*.

Corollary 4 *Let X be an EBP process. Observe X at some time t 'uniformly' distributed over $[0, \infty)$, and let k be such that $X(t) \in C^n_k$. Then $Z^{n+1}_{\kappa(n,n+1,k)}$ has distribution $xp(x)/\mu$ and S^n_k is uniformly distributed over $\{1, \ldots, Z^{n+1}_{\kappa(n,n+1,k)}\}$.*

If $S^n_k = Z^{n+1}_{\kappa(n,n+1,k)}$ then $\alpha^n_k = 1^+$ or -1^- with equal probability. Otherwise if S^n_k is odd then $\alpha^n_k = 0^+$ or 0^- with equal probability. If S^n_k is even, $S^n_k \neq Z^{n+1}_{\kappa(n,n+1,k)}$, then $\alpha^n_k = 1^-$ or -1^+ with equal probability.

Finally, the sampling distributions of S_k^n and $Z_{\kappa(n,n+1,k)}^{n+1}$ are independent of the length of C_k^n.

Our simulation algorithm uses a modified crossing state. For some $n_{\max} = n_{\max}(k)$

$$\overline{\mathcal{X}}^m(k) = \{\mathcal{X}^{m,m}(k), \ldots, \mathcal{X}^{m,n_{\max}}(k)\}.$$

We give a procedure `Expand` to increase $n_{\max}$ when necessary. Let $\bar{p}$ be the distribution given by $\bar{p}(n) = np(n)/\mu$.

`Procedure Expand` $\overline{\mathcal{X}}^m(k)$
    `While` $S^{m,n_{\max}}(k) = Z^{m,n_{\max}+1}(k)$ `Do`
        $\kappa(m, n_{\max}+1, k) = 1$
        Generate $Z^{m,n_{\max}+2}(k)$ using distribution $\bar{p}$
        Generate $S^{m,n_{\max}+1}(k) \sim U\{1, \ldots, Z^{m,n_{\max}+2}(k)\}$
        `If` $\alpha^{m,n_{\max}}(k) = 1^+$ `Then`
            `If` $S^{m,n_{\max}+1}(k) = Z^{m,n_{\max}+2}(k)$ `Then`
                $\alpha^{m,n_{\max}+1}(k) = 1^+$
            `Else If` $S^{m,n_{\max}+1}(k)$ is odd `Then`
                $\alpha^{m,n_{\max}+1}(k) = 0^+$
            `Else`
                $\alpha^{m,n_{\max}+1}(k) = -1^+$
            `End If`
        `Else` $(\alpha^{m,n_{\max}}(k) = -1^-)$
            `If` $S^{m,n_{\max}+1}(k) = Z^{m,n_{\max}+2}(k)$ `Then`
                $\alpha^{m,n_{\max}+1}(k) = -1^-$
            `Else If` $S^{m,n_{\max}+1}(k)$ is odd `Then`
                $\alpha^{m,n_{\max}+1}(k) = 0^-$
            `Else`
                $\alpha^{m,n_{\max}+1}(k) = 1^-$
            `End If`
        `End If`
        $n_{\max} = n_{\max} + 1$
    `End While`
`End Procedure`

We will also use procedure `Increment` with two changes. Firstly the line (†) is changed to

$$\mathcal{X}^{m,q}(k+1) = \mathcal{X}^{m,q}(k) \ \texttt{for} \ q = n+1, \ldots, n_{\max}$$

Secondly, at the point (‡) we assign types differently if $n = n_{\max}$, as in this case we can not use $\alpha^{m,n+1}(k+1)$ to determine $\alpha^{m,n}(k+1)$. Instead we determine $\alpha^{m,n}(k+1)$ randomly.

    `If` $S^n_{\kappa(m,n,k+1)} = Z^{n+1}_{\kappa(m,n+1,k+1)}$ `Then`
        `If` $\alpha^n_{\kappa(m,n,k)} = 0^+$ `Then`
            $\alpha^n_{\kappa(m,n,k+1)} = 1^+$
        `Else` $(\alpha^n_{\kappa(m,n,k)} = 0^-)$
            $\alpha^n_{\kappa(m,n,k+1)} = -1^-$
        `End If`

```
Else If S^n_{κ(m,n,k+1)} is odd Then
    α^n_{κ(m,n,k+1)} = 0^+ or 0^- with equal probability
Else
    If α^n_{κ(m,n,k)} = 0^+ Then α^n_{κ(m,n,k+1)} = 1^- End If
    If α^n_{κ(m,n,k)} = 0^- Then α^n_{κ(m,n,k+1)} = -1^+ End If
End If
```

We can now give our simulation algorithm.

```
Procedure Simulate
    (Given X̄^m(k) and X^m(k+1) returns X̄^m(k+1) and X^m(k+2))
    Expand X̄^m(k)
    Increment X^{m,m}(k)
    If α^{m,m}(k+1) = i^+ Then
        X^m(k+2) = X^m(k+1) + 2^m
    Else
        X^m(k+2) = X^m(k+1) - 2^m
    End If
End Procedure
```

To initialise the crossing state we have the following

```
Procedure Initialise X̄^m
    n_max = m
    κ(m,m,1) = 1
    Generate Z^{m,m+1}(1) using distribution p̄
    Generate S^{m,m}(1) ~ U{1,...,Z^{m,m+1}(1)}
    If S^{m,m}(1) = Z^{m,m+1}(1) Then
        α^{m,m}(1) = 1^+ or -1^- with equal probability
    Else If S^{m,m}(1) is odd Then
        α^{m,m}(1) = 0^+ or 0^- with equal probability
    Else
        α^{m,m}(1) = 1^- or -1^+ with equal probability
    End If
End Procedure
```

We assume we are given $X^m(1)$ (recall we take $k = 1$ as our starting time). Given $\overline{\mathcal{X}}^m(1)$ and $X^m(1)$ we get $X^m(2) = X^m(1) \pm 2^m$ according to $\alpha^{m,m}(1) = i^{\pm}$.

3.1 Performance

On average, X^m starts a new level n crossing every μ^{n-m} steps. It follows immediately that $n_{\max}(k) = O(\log k)$. To generate a new step of X^m it is only necessary to store the previous value and $\overline{\mathcal{X}}^m$. Thus to generate n steps we require $O(\log n)$ storage.

The expected operation count of procedure `Expand` is finite and independent of $n_{\max}(k)$. The operation count of procedure `Increment` is proportional to $n_{\max}(k)$. Thus the number of operations used by `Simulate` to generate n steps of X^m is of order

$$\sum_{k=1}^{n} \log k = \log n! \approx \log(\sqrt{2\pi n} e^{-n} n^n) = O(n \log n).$$

(Using Stirling's formula for the approximation.)

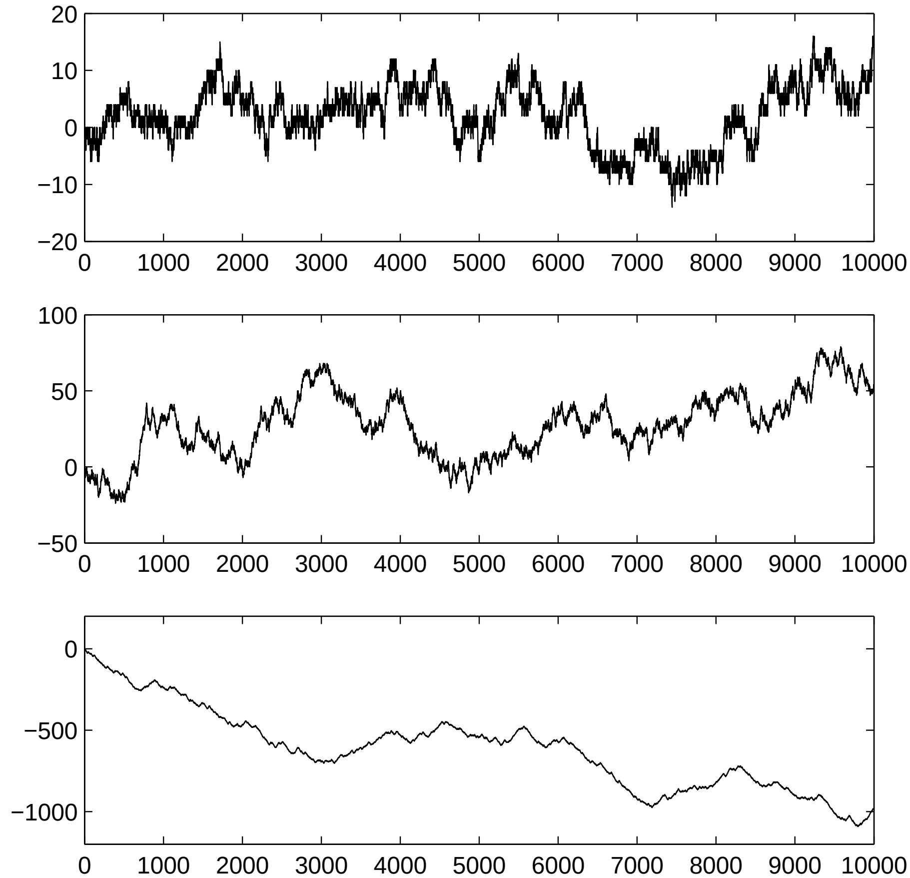

Figure 2: Self-similar traces with $H = 0.3010$, 0.5 and 0.7565 respectively.

3.2 Crossing times

It is easily seen that the level m crossing times of the EBP process X are independent and have the same distribution as the normed limit W of the Galton-Watson branching process with offspring distribution p (up to some constant scaling). Thus, to simulate X at spatial resolution 2^m we simulate X^m and use i.i.d. level m crossing times for the times between jumps. It is possible to sample from the distribution of W approximately, by simulating a finite number of generations and normalising by the expected population size.

In practice if m is small, then a rough approximation to the distribution of W is sufficient (even a constant approximation is m is small enough). It is possible to sample from the distribution of W with high accuracy very efficiently, but as this is really ancillary to the principal content of this paper we will not consider this problem further here.

3.3 Examples

We illustrate the algorithm with some simulated traces of EBP process. In Figure 2 we use the offspring distribution $p(2k) = a(1-a)^{k-1}$ for $a \in (0,1)$. This gives $\mu = 2/a$ and $H = \log 2/(\log 2 - \log a)$. From top to bottom we have $(a, H) = (0.2, 0.3010)$, $(0.5, 0.5)$ and $(0.8, 0.7565)$.

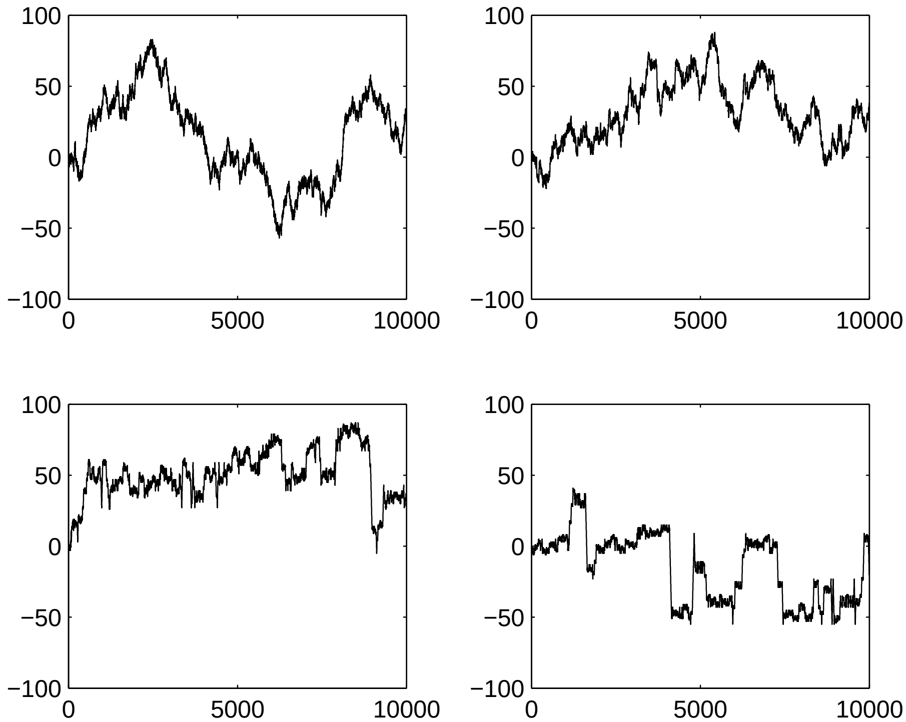

Figure 3: Self-similar traces with $H = 0.5$ but different offspring distributions.

In Figure 3 we have four processes with the same H value of 0.5. The offspring distributions for each process are in order

Top left:	$p(2) = 0.5,$	$p(6) = 0.5;$
Top right:	$p(2) = 0.75,$	$p(10) = 0.25;$
Bottom left:	$p(2) = 0.9,$	$p(22) = 0.1;$
Bottom right:	$p(2) = 0.95,$	$p(42) = 0.05.$

4 Proofs

Proof of Theorem 1 We construct a crossing of size 1. Given the self-similarity of the process, this can be scaled to give a sample of arbitrary length. The method we use dates back to Knight [6] and Barlow & Perkins [7].

We first define a number of ancillary processes. For $m \leq 0$ let Y^m be a random walk with steps of size 2^m at times $\mu^m \mathbb{Z}_+$. Put $Y^0(0) = 0$ and $Y^0(1) = 1$, then construct Y^{m-1} from Y^m by replacing step k of Y^m by a sequence of Z_k^m steps of size 2^{m-1}, where $\mathrm{P}(Z_k^m = x) = p(x)$. These are the level $m-1$ sub-crossings of crossing k at level m. The Z_k^m are independent and identically distributed.

Each crossing can be up or down. A sequence of Z_k^m sub-crossings consists of $(Z_k^m - 2)/2$ excursions followed by a direct crossing. An excursion is an up-down or a down-up pair; a direct crossing is an up-up or a down-down pair. If the parent crossing is up, then the sub-crossings end up-up, otherwise they end down-down. We allow each excursion to be up-down or down-up with equal probability (though note that other choices are possible).

We extend Y^m from $\mu^m\mathbb{Z}_+ \to 2^m\mathbb{Z}$ to $\mathbb{R}_+ \to \mathbb{R}$ by linear interpolation. Also let $T^m = \inf\{t : Y^m(t) = 1\}$, and for $t > T^m$ put $Y^m(t) = 1$. The interpolated Y^m has continuous sample paths. We will show that with probability 1, as $m \to -\infty$ the sample path of Y^m converges uniformly on any finite interval. The limiting sample path is thus continuous.

For $n \le m$ let $T_0^{m,n} = 0$ and $T_{k+1}^{m,n} = \inf\{t > T_k^{m,n} : Y^n(t) \in 2^m\mathbb{Z}, Y^n(t) \ne Y^n(T_k^{m,n})\}$. If $Y^n(T_k^{m,n}) = 1$ then we put $T_{k+1}^{m,n} = \infty$. The $T_k^{m,n}$ are the level m hitting times of Y^n. The k-th level m crossing time of Y^n is $W_k^{m,n} = T_k^{m,n} - T_{k-1}^{m,n}$. For each m and k, $\{\mu^{m-n}W_k^{m,n}\}_{n=m}^{-\infty}$ is a Galton-Watson branching process. Thus since p is regular there exist i.i.d. continuous non-negative r.v.s W_k^m with mean 1 such that (see for example Athreya & Ney [8])

$$W_k^{m,n} \to W_k^m \text{ with probability 1.}$$

Let $T_k^m = \sum_{j=1}^k W_k^m = \lim_{n\to-\infty} T_k^{m,n}$.

Fix $\epsilon, \delta > 0$ and $T > 0$. We will find a u such that for all $r, s \le u$ and $t \in [0,T]$

$$|Y^r(t) - Y^s(t)| \le \epsilon \text{ with probability } 1-\delta \tag{2}$$

Given $t \in [0,T]$, let $k = k(n,t)$ be such that

$$T_{k-1}^n \le t < T_k^n$$

then for any $r, s \le n$

$$\begin{aligned} &|Y^r(t) - Y^s(t)| \\ &\le |Y^r(t) - Y^r(T_k^{n,r})| + |Y^r(T_k^{n,r}) - Y^s(T_k^{n,s})| + |Y^s(T_k^{n,s}) - Y^s(t)| \\ &= |Y^r(t) - Y^r(T_k^{n,r})| + |Y^s(T_k^{n,s}) - Y^s(t)| \end{aligned} \tag{3}$$

noting that $Y^r(T_k^{n,r}) = Y^s(T_k^{n,s}) = Y^n(k\mu^n)$. Now, let $j = j(u,T)$ be the smallest j such that

$$T_j^{n,u} > T$$

then as $u \to -\infty$, $j(u,T) \to j(T) < \infty$ a.s., so we can choose a u such that for all $q \le u$

$$\max_{i\le j}\{|T_i^{n,q} - T_i^n|\} < \min_{i\le j} W_i^n \text{ with probability } 1-\delta.$$

Thus for any $q \le p$

$$T_{k-2}^{n,q} < t < T_{k+1}^{n,q}$$

and

$$|Y^q(t) - Y^q(T_k^{n,q})| = |Y^q(t) - Y^n(k\mu^n)| \le 3\cdot 2^n$$

since $Y^q(T_{k-2}^{n,q}) = Y^n((k-2)\mu^n)$, $Y^q(T_{k+1}^{n,q}) = Y^n((k+1)\mu^n)$ and in three steps Y^n can move at most distance $3\cdot 2^n$. Applying this to (3) proves (2), taking n small enough that $6\cdot 2^n \le \epsilon$. Thus as ϵ and δ are arbitrary, Y^n converges to some (necessarily continuous) Y uniformly on all closed intervals $[0,T]$, with probability 1.

By construction $Y(T_k^m) = Y^m(k\mu^m)$ for all m and k. The self-similarity (1) is also a direct consequence of the construction. □

Proof of Lemma 3 Let $T_k = \sum_{i=1}^k \sum_{j=1}^{n(i)} X_{i,j}$ and let $\mathcal{P}_k$ be the partition of $[0,T_k)$ given by $X_{1,1},\ldots,X_{k,N(k)}$. Given $\mathcal{P}_k$, choose x uniformly on $[0,T_k)$ and let X_k^* and N_k^* be the interval length and family size of x. Let $S_k(n) = \#\{i : 1 \le i \le k,\, N(i) = n\}$ then sending $k \to \infty$

$$
\begin{aligned}
&\mathrm{P}(N_k^* = n,\, J^* = l,\, X_k^* \le x \,|\, \mathcal{P}) \\
&= \sum_{i=1}^{k} \sum_{j=1}^{N(i)} \mathrm{P}(N_k^* = n,\, J^* = l,\, X_k^* \le x \,|\, N_k^* = N(i),\, X_k^* = X_{i,j},\, \mathcal{P}) \frac{X_{i,j}}{T_k} \\
&= \sum_{i=1}^{k} \sum_{j=1}^{n} I_{\{n\}}(N(i)) I_{\{l\}}(j) I_{[0,x]}(X_{i,j}) \frac{X_{i,j}}{T_k} \\
&= \frac{k}{T_k} \frac{S_k(n)}{k} \frac{1}{S_k(n)} \sum_{i=1}^{k} I_{\{n\}}(N(i)) X_{i,l} I_{[0,x]}(X_{i,l}) \\
&\to \frac{1}{\mu m} p(n) \mathrm{E} X_{i,j} I_{[0,x]}(X_{i,j}) \text{ with probability 1} \\
&= \frac{np(n)}{\mu} \frac{1}{n} \int_0^x \frac{y}{m} dF(y).
\end{aligned}
$$

Differentiating w.r.t. x gives the result.

By integrating/summing out the other terms, one can easily show that the marginal distributions of N^* and X^* are given by

$$
\begin{aligned}
\mathrm{P}(N^* = n) &= \frac{np(n)}{\mu}; \\
\mathrm{P}(x \le X^* < x + dx) &= \frac{x}{m} dF(x).
\end{aligned}
$$

Similarly, the conditional distribution of J^* given N^* is given by

$$
\mathrm{P}(J^* = l \,|\, N^* = n) = \frac{1}{n} \text{ for } 1 \le l \le n.
$$

□

References

[1] J.R.M. Hosking. Modelling persistence is hydrological time series using fractional differencing. *Water Resources Research*, 20:1898–1908, 1984.

[2] A.T.A. Wood and G. Chan. Simulation of stationary Gaussian processes in $[0,1]^d$. *J. Computational Graphical Stat.*, 3:409–432, 1994.

[3] V. Paxson. Fast, approximate synthesis of fractional gaussian noise for generating self-similar network traffic. *Comp. Comm. Rev.*, 27:5–18, 1997.

[4] D.R. Cox. Long range dependence: A review. In *Statistics: An Appraisal*, David, H.A. & David, H.T., eds. Iowa State University Press, 1984.

[5] O.D. Jones and Y. Shen. Estimating the Hurst index of a self-similar process via the crossing tree. *Signal Processing Letters*, 11:416–419, 2004.

[6] F.B. Knight. *Essentials of Brownian Motion and Diffusion*. Number 18 in AMS Math. Surveys. 1981.

[7] M.T. Barlow and E.A. Perkins. Brownian motion on the Sierpinski gasket. *Prob. Th. Rel. Fields*, 79:543–623, 1988.

[8] K.B. Athreya and P.E. Ney. *Branching Processes*. Springer-Verlag, 1972.