

Fast exact simulation of the first passage event of a tempered stable subordinator (Part 1)

Jorge González Cázares



Feng Lin



Aleksandar Mijatović



EPSRC

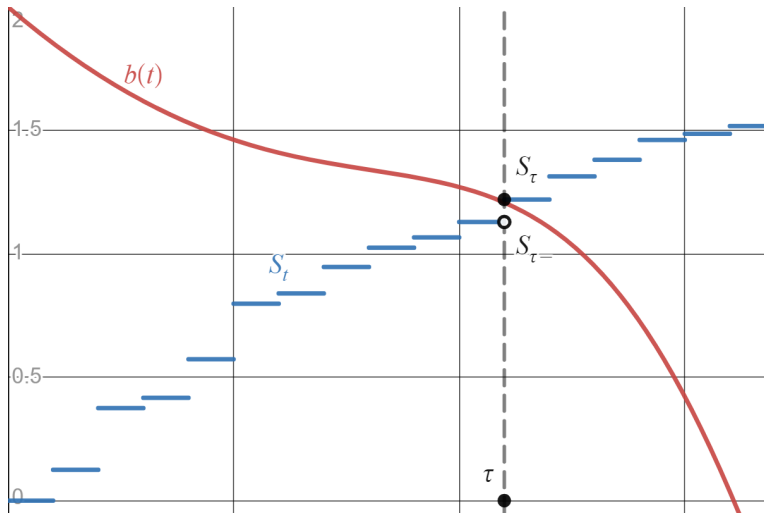
Engineering and Physical Sciences
Research Council

University of Warwick & The Alan Turing Institute

EP/W006227/1 & EP/V009478/1

March 2022

The problem: first passage across a non-increasing curve



Basic idea (I)

Fix any $\theta > 0$ and $\alpha \in (0, 1)$. Let S be a driftless stable subordinator under the probability measure $\mathbb{P}$ with Lévy measure $\nu(dx) = w_\alpha^{-1} \theta x^{-\alpha-1} dx$ where $w_\alpha = \Gamma(1 - \alpha)/\alpha$. Denote by $(\mathcal{G}_t)_{t \geq 0}$ be the right-continuous filtration generated by S . Given $q \geq 0$, let $\mathbb{P}_q$ be the probability measure with Radon–Nikodym derivative

$$\left. \frac{d\mathbb{P}_q}{d\mathbb{P}} \right|_{\mathcal{G}_t} = \exp \left(-qS_t + q^\alpha \theta t \right), \quad t > 0.$$

Then S is a tempered stable subordinator under $\mathbb{P}_q$ with Lévy measure $\nu_q(dx) = w_\alpha^{-1} \theta e^{-qx} x^{-\alpha-1} dx$.

Basic idea (II)

Proposition 1

For any $B \in \mathcal{G}_t = \sigma(S_u : u \in [0, t])$, let $E \sim \text{Exp}(1)$ and $E \perp\!\!\!\perp \mathcal{G}_t$. Then

$$\mathbb{P}_q[B] = \mathbb{P}[B | E \geq qS_t]. \quad (2.1)$$

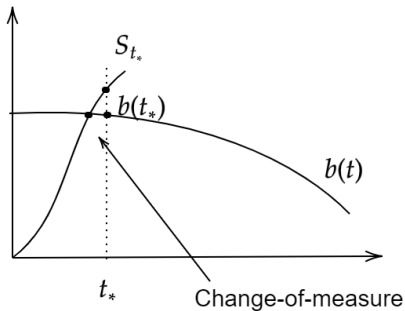
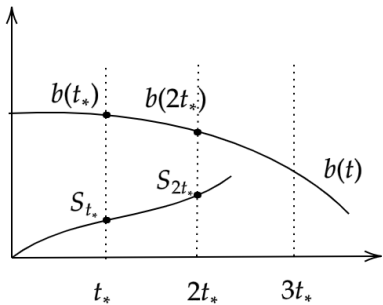
Moreover, for any $C \in \mathcal{G}_t$ satisfying $\mathbb{P}[C] > 0$, we have

$$\mathbb{P}_q[B | C] = \mathbb{P}[B | C, E \geq qS_t]. \quad (2.2)$$

Thus, it is natural to expect that the simulation of the first passage triplet $(\tau_b, S_{\tau_b-}, S_{\tau_b})$ under $\mathbb{P}_q$ can essentially be reduced to the case $q = 0$ on the stochastic interval $[0, \tau_b]$.

Time grid

Make use of the change-of-measure idea by introducing a time grid and sampling S on the grid until the interval in which τ_b lies is identified!



First main algorithm

Algorithm 1 Simulation of the triplet $(\tau_b, S_{\tau_b-}, S_{\tau_b})$ under $\mathbb{P}_q$

Require: Parameters $\alpha \in (0, 1)$, $\theta > 0$, $q \geq 0$ and function $b : [0, \infty) \rightarrow [0, \infty)$

- 1: Set $t_* \leftarrow (2qb(0) + 1 - 2^{-\alpha})/((2^\alpha - 1)q^\alpha\theta)$, $T \leftarrow 0$ and $U \leftarrow 0$
- 2: **repeat**
- 3: Sample s from the law of S_{t_*} under $\mathbb{P}_q$
- 4: **if** $s < b(t_*)$ **then**
- 5: Update $T \leftarrow T + t_*$, $U \leftarrow U + s$ and $b(\cdot) \leftarrow b(\cdot + t_*) - s$
- 6: **end if**
- 7: **until** $s \geq b(t_*)$
- 8: **repeat**
- 9: Sample (t, u, v) from the law of $(\tau_b, S_{\tau_b-}, S_{\tau_b})|_{\{\tau_b \leq t_*\}}$ under $\mathbb{P}$
- 10: Sample w from the law of S_{t_*-t} under $\mathbb{P}$ and $E' \sim \text{Exp}(1)$
- 11: **until** $E' \geq q(w + v)$
- 12: **return** $(T + t, U + u, U + v)$

First simulation method

Theorem 1

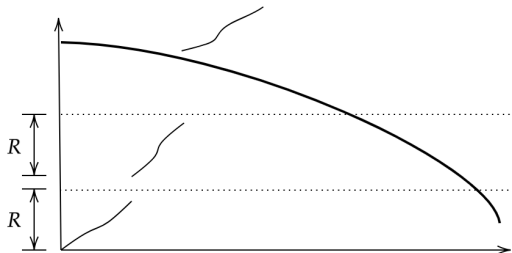
Algorithm 1 samples from the law of the triplet $(\tau_b, S_{\tau_b-}, S_{\tau_b})$ of a tempered stable subordinator S under $\mathbb{P}_q$. Moreover, the expected running time is bounded by

$$\kappa_1 e^{2q b(0)/(2^\alpha - 1)} (1 + (1 - \alpha)^{-3} + |\log \alpha| + \log N),$$

where the constant κ_1 does not depend on the parameters $\alpha \in (0, 1)$, $\theta \in (0, \infty)$ or $q \in [0, \infty)$, and N specifies the required number of precision bits of the output.

Improvement

The upper bound on the running time of Algorithm 1 has a term of order $\mathcal{O}(\exp(2qb(0)/(2^\alpha - 1)))$, which may be quite large if either $b(0)$, q or $1/\alpha$ are large. However, the exponential dependence on these parameters of the complexity of Algorithm 1 can be easily turned into a linear dependence by picking $R > 0$ and applying Algorithm 1 successively with the boundary function $t \mapsto \min\{b(t), R\}$.



Improvement: new complexity

Corollary 1

The improved version of Algorithm 1 samples from the law of $(\tau_b, S_{\tau_b-}, S_{\tau_b})$ under $\mathbb{P}_q$ with expected running time bounded by

$$\kappa_2(1 + qb(0)/\alpha)((1 - \alpha)^{-3} + |\log \alpha| + \log N)/\alpha,$$

where κ_2 depends on neither $\alpha \in (0, 1)$, $\theta \in (0, \infty)$ nor $q \in [0, \infty)$, and N specifies the required number of precision bits of the output.

First passage event for stable subordinator

Proposition 2 (Chi, 2016)

Let S be a stable subordinator and $b : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ a non-increasing absolutely continuous function with $b(0) > 0$. Define $T_b := \inf\{t > 0 : b(t) = 0\}$. The following statements hold.

First passage event for stable subordinator

Proposition 2 (Chi, 2016)

Let S be a stable subordinator and $b : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ a non-increasing absolutely continuous function with $b(0) > 0$. Define $T_b := \inf\{t > 0 : b(t) = 0\}$. The following statements hold.

- (a) $\tau_b \stackrel{d}{=} B^{-1}(S_1)$, where $B(t) := t^{-1/\alpha} b(t)$ is a strictly decreasing continuous function $B : (0, T_b) \rightarrow (0, \infty)$, with inverse $B^{-1} : (0, \infty) \rightarrow (0, T_b)$. Hence, $\mathbb{P}(0 < \tau_b < T_b) = 1$.

First passage event for stable subordinator

Proposition 2 (Chi, 2016)

Let S be a stable subordinator and $b : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ a non-increasing absolutely continuous function with $b(0) > 0$. Define $T_b := \inf\{t > 0 : b(t) = 0\}$. The following statements hold.

- (a) $\tau_b \stackrel{d}{=} B^{-1}(S_1)$, where $B(t) := t^{-1/\alpha} b(t)$ is a strictly decreasing continuous function $B : (0, T_b) \rightarrow (0, \infty)$, with inverse $B^{-1} : (0, \infty) \rightarrow (0, T_b)$. Hence, $\mathbb{P}(0 < \tau_b < T_b) = 1$.
- (b) The probability of S creeping at b , conditional on at time $\tau_b = t \in (0, T_b)$, equals
- $$\mathbb{P}[S(\tau_b) = b(\tau_b) | \tau_b = t] = \mathbb{P}[\Delta_S(\tau_b) = 0 | \tau_b = t] = -b'(t)/(-b'(t) + \alpha^{-1} t^{-1} b(t)).$$

First passage event for stable subordinator

Proposition 2 (Chi, 2016)

Let S be a stable subordinator and $b : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ a non-increasing absolutely continuous function with $b(0) > 0$. Define $T_b := \inf\{t > 0 : b(t) = 0\}$. The following statements hold.

(a) $\tau_b \stackrel{d}{=} B^{-1}(S_1)$, where $B(t) := t^{-1/\alpha} b(t)$ is a strictly decreasing continuous function $B : (0, T_b) \rightarrow (0, \infty)$, with inverse $B^{-1} : (0, \infty) \rightarrow (0, T_b)$. Hence, $\mathbb{P}(0 < \tau_b < T_b) = 1$.

(b) The probability of S creeping at b , conditional on at time $\tau_b = t \in (0, T_b)$, equals
$$\mathbb{P}[S(\tau_b) = b(\tau_b) | \tau_b = t] = \mathbb{P}[\Delta_S(\tau_b) = 0 | \tau_b = t] = -b'(t) / (-b'(t) + \alpha^{-1} t^{-1} b(t)).$$

(c) Conditional on $A = \{\Delta_S(\tau_b) > 0\}$ and $\tau_b = t \in (0, T_b)$, we have

$$\mathbb{P}[S_{\tau_b-} \in du | \Delta_S(\tau_b) > 0, \tau_b = t] = \frac{w_{\alpha}^{-1} \theta t}{b(t) g_t(b(t))} (b(t) - u)^{-\alpha} g_t(u) du,$$

$$\mathbb{P}[\Delta_S(\tau_b) \leq v | \Delta_S(\tau_b) > 0, \tau_b = t, S_{\tau_b-} = u] = 1 - ((b(t) - u)/v)^{\alpha},$$

for $u \in (0, b(t))$ and $v \in (b(t) - u, \infty)$.

Simulation algorithm for stable subordinator

First sample the crossing time τ_b and determine whether the crossing occurred via a jump $\Delta_S(\tau_b) > 0$ or by creeping $\Delta_S(\tau_b) = 0$. In the latter case, return $(\tau_b, b(\tau_b), b(\tau_b))$, in the former, sample the **undershoot** and the **overshoot**.

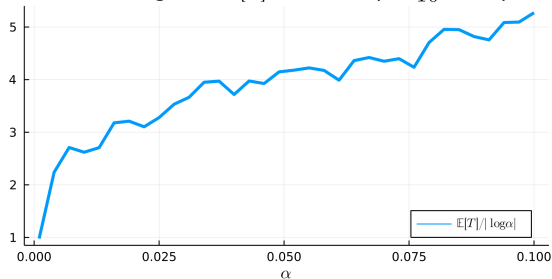
Algorithm 2 Simulation of the triplet $(\tau_b, S_{\tau_b-}, S_{\tau_b})$ under $\mathbb{P}$

Require: Parameters $\alpha \in (0, 1)$, $\theta > 0$ and functions $b : [0, \infty) \rightarrow [0, \infty)$ and $B^{-1} : (0, \infty) \rightarrow (0, T_b)$ specified in Proposition 2

- 1: Generate $U_1 \sim U(0, 1)$ and a stable variable $V_1 \sim \mathcal{L}(S_1)$ and set $t \leftarrow B^{-1}(V_1)$
- 2: **if** $U_1 \leq -b'(t)/(-b'(t) + \alpha^{-1}t^{-1}b(t))$ **then**
- 3: **return** $(t, b(t), b(t))$
- 4: **else**
- 5: Generate $U_2 \sim U(0, 1)$ and V_2 from the undershoot law of $S_{t-} | \{\tau_b = t, b(\tau_b) = w, \Delta_S(\tau_b) > 0\}$
- 6: **return** $(t, V_2, V_2 + (b(t) - V_2)U_2^{-1/\alpha})$
- 7: **end if**

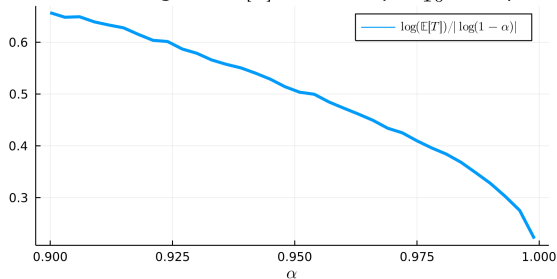
Complexities

Mean running time $\mathbb{E}[T]$ (seconds per 10^4 samples)



(a) $\mathbb{E}[T] = \mathcal{O}(|\log \alpha|)$ as $\alpha \downarrow 0$. The picture confirms this (the reported value of $\mathbb{E}[T]$ for $\alpha = 0.001$ is 9.30 seconds per 10^4 samples).

Mean running time $\mathbb{E}[T]$ (seconds per 10^4 samples)



(b) $\mathbb{E}[T] = \mathcal{O}((1 - \alpha)^{-3})$ as $\alpha \uparrow 1$. The picture confirms this (the reported value of $\mathbb{E}[T]$ for $\alpha = 0.999$ is 6.28 seconds per 10^4 samples).

Theoretical bound on the complexity:

$$\kappa_2 \left((1 - \alpha)^{-3} + |\log \alpha| + \log N \right).$$

Simulation algorithm for stable subordinator: the ideas

- Although the density of the undershoot S_{τ_b-} admits an explicit form in Proposition 2, it is unbounded and does not lend itself well to direct efficient accept-reject methods.

Simulation algorithm for stable subordinator: the ideas

- Although the density of the undershoot S_{τ_b-} admits an explicit form in Proposition 2, it is unbounded and does not lend itself well to direct efficient accept-reject methods.
- Strategy: break up the domain into regions, depending on the values of the possibly random parameters, and use a mix of direct accept-reject methods and **Devroye's algorithm** for log-concave densities

Simulation algorithm for stable subordinator: the ideas

- Although the density of the undershoot S_{τ_b-} admits an explicit form in Proposition 2, it is unbounded and does not lend itself well to direct efficient accept-reject methods.
- Strategy: break up the domain into regions, depending on the values of the possibly random parameters, and use a mix of direct accept-reject methods and **Devroye's algorithm** for log-concave densities
- **Devroye's algorithm** plays a key role precisely in the regions where the density of S_{τ_b-} decreases sharply to *super-exponentially small* values

Simulation algorithm for stable subordinator: the ideas

- Although the density of the undershoot S_{τ_b-} admits an explicit form in Proposition 2, it is unbounded and does not lend itself well to direct efficient accept-reject methods.
- Strategy: break up the domain into regions, depending on the values of the possibly random parameters, and use a mix of direct accept-reject methods and **Devroye's algorithm** for log-concave densities
- **Devroye's algorithm** plays a key role precisely in the regions where the density of S_{τ_b-} decreases sharply to *super-exponentially small* values
- Our algorithms combine (I) binary search (bisection method), possibly followed by Newton–Raphson (rigorous verification of quadratic convergence) and (II) accept-reject. For both, we do a careful complexity analysis.

Simulation algorithm for stable subordinator: the ideas

- Although the density of the undershoot S_{τ_b-} admits an explicit form in Proposition 2, it is unbounded and does not lend itself well to direct efficient accept-reject methods.
- Strategy: break up the domain into regions, depending on the values of the possibly random parameters, and use a mix of direct accept-reject methods and **Devroye's algorithm** for log-concave densities
- **Devroye's algorithm** plays a key role precisely in the regions where the density of S_{τ_b-} decreases sharply to *super-exponentially small* values
- Our algorithms combine (I) binary search (bisection method), possibly followed by Newton–Raphson (rigorous verification of quadratic convergence) and (II) accept-reject. For both, we do a careful complexity analysis. **This is crucial because parameter values themselves are often random!**

Simulation algorithm for stable subordinator: the ideas

- Although the density of the undershoot S_{τ_b-} admits an explicit form in Proposition 2, it is unbounded and does not lend itself well to direct efficient accept-reject methods.
- Strategy: break up the domain into regions, depending on the values of the possibly random parameters, and use a mix of direct accept-reject methods and **Devroye's algorithm** for log-concave densities
- **Devroye's algorithm** plays a key role precisely in the regions where the density of S_{τ_b-} decreases sharply to *super-exponentially small* values
- Our algorithms combine (I) binary search (bisection method), possibly followed by Newton–Raphson (rigorous verification of quadratic convergence) and (II) accept-reject. For both, we do a careful complexity analysis. **This is crucial because parameter values themselves are often random! (Note that the infinite expected running time in Chi's algorithm arises essentially due to an accept-reject step having a random acceptance probability.)**

Simulation of the undershoot of a stable subordinator

A key non-trivial component in the previous algorithm is the simulation of the **undershoot**. Proposition 2(c) gives the density of the undershoot of S_t given $\tau_b = t$ and $b(\tau_b) = w$.

Simulation of the undershoot of a stable subordinator

A key non-trivial component in the previous algorithm is the simulation of the **undershoot**.

Proposition 2(c) gives the density of the undershoot of S_t given $\tau_b = t$ and $b(\tau_b) = w$.

1. Use the integral representation of the stable law in terms of an explicit convex function σ_α .

Thus, we sample (ζ_s, Y_s) (where $s = (\theta t)^{-1/\alpha} w$) from the density

$$\psi_s(x, y) \propto (s - x^{-1/r})^{-\alpha} \sigma_\alpha(y) e^{-\sigma_\alpha(y)x}, \quad (x, y) \in (s^{-r}, \infty) \times (0, 1),$$

(where $r = \frac{\alpha}{1-\alpha}$) and return $(\theta t)^{1/\alpha} \zeta_s^{-1/r}$.

Simulation of the undershoot of a stable subordinator

A key non-trivial component in the previous algorithm is the simulation of the **undershoot**.

Proposition 2(c) gives the density of the undershoot of S_t given $\tau_b = t$ and $b(\tau_b) = w$.

1. Use the integral representation of the stable law in terms of an explicit convex function σ_α .

Thus, we sample (ζ_s, Y_s) (where $s = (\theta t)^{-1/\alpha} w$) from the density

$$\psi_s(x, y) \propto (s - x^{-1/r})^{-\alpha} \sigma_\alpha(y) e^{-\sigma_\alpha(y)x}, \quad (x, y) \in (s^{-r}, \infty) \times (0, 1),$$

(where $r = \frac{\alpha}{1-\alpha}$) and return $(\theta t)^{1/\alpha} \zeta_s^{-1/r}$.

2. Dominate ψ_s by a mixture of densities that can both be simulated using rejection sampling:

$$\tilde{\psi}_s^{(1)}(x, y) \propto \sigma_\alpha(y) e^{-\sigma_\alpha(y)x} \quad \text{and} \quad \tilde{\psi}_s^{(2)}(x, y) \propto \sigma_\alpha(y) e^{-\sigma_\alpha(y)x} (x - s^{-r})^{-\alpha}$$

satisfy $\tilde{\psi}_s := p \tilde{\psi}_s^{(1)} + (1 - p) \tilde{\psi}_s^{(2)} \geq c_\psi \psi_s$ for some explicit $p, c_\psi \in (0, 1)$.

Simulation of the undershoot of a stable subordinator

A key non-trivial component in the previous algorithm is the simulation of the **undershoot**.

Proposition 2(c) gives the density of the undershoot of S_t given $\tau_b = t$ and $b(\tau_b) = w$.

1. Use the integral representation of the stable law in terms of an explicit convex function σ_α .

Thus, we sample (ζ_s, Y_s) (where $s = (\theta t)^{-1/\alpha} w$) from the density

$$\psi_s(x, y) \propto (s - x^{-1/r})^{-\alpha} \sigma_\alpha(y) e^{-\sigma_\alpha(y)x}, \quad (x, y) \in (s^{-r}, \infty) \times (0, 1),$$

(where $r = \frac{\alpha}{1-\alpha}$) and return $(\theta t)^{1/\alpha} \zeta_s^{-1/r}$.

2. Dominate ψ_s by a mixture of densities that can both be simulated using rejection sampling:

$$\tilde{\psi}_s^{(1)}(x, y) \propto \sigma_\alpha(y) e^{-\sigma_\alpha(y)x} \quad \text{and} \quad \tilde{\psi}_s^{(2)}(x, y) \propto \sigma_\alpha(y) e^{-\sigma_\alpha(y)x} (x - s^{-r})^{-\alpha}$$

satisfy $\tilde{\psi}_s := p \tilde{\psi}_s^{(1)} + (1 - p) \tilde{\psi}_s^{(2)} \geq c_\psi \psi_s$ for some explicit $p, c_\psi \in (0, 1)$.

3. Sample from $\tilde{\psi}_s^{(1)}$ using **Devroye's algorithm** for log-concave densities.
Sampling from $\tilde{\psi}_s^{(2)}$ efficiently is hard! Split domain, break into cases and find dominating densities to sample from.