Simulations of Lévy Processes

Research Paper

by

Yifei Deng

Supervisor: Professor Christiane Lemieux

A project
In partial fulfilment of the
requirement for
Master of Quantitative Finance

Department of Statistics and Acturial Science

University of Waterloo

January 14, 2021

Acknowledgements

I would like to thank Professor Christiane Lemieux who gives me suggestion and encouragements throughout the research and made this research paper possible.

Table of Contents

Chapter 1

Introduction

The Variance Gamma (VG) process, a pure jump Lévy process with finite variations in an infinite time horizon, has been known for modelling the uncertain underlying stock market returns in the financial literature for several decades. The VG model was first formally introduced by Madan and Seneta (1990) [14] as an extension of the geometric Brownian motion with two additional parameters for adjusting the skewness and kurtosis of the distribution of the underlying stock return. The VG model resolves many problems that the classical Black-Scholes Merton diffusion model [6] is incapable of solving, including alleviating the volatility smile effect, fitting fat tails of the stock returns distribution and tracking large market movements, according to Cont and Tankov (2004)[7]. Later, to accommodate the VG model into a risk-neutral pricing framework, Madan and Seneta (1998)[13] extend the VG model as a continuous-time model for a log-price underlying and a closed-form solution for European vanilla options is presented. More recently, the option pricing under the VG model has been applied to the American type of options in Hirsa and Madan (2004)[1]. In order to speed up the sampling of the VG process in option pricing, Avramidis and L'Ecuyer (2006) [2] have developed the difference-of-gamma sampling (DGBS) in Monte-Carlo (MC) and Randomized Quasi-Monte Carlo (rQMC) simulations that gives estimated exotic option values efficiently. This DGBS algorithm is further advanced by Becker (2010) [4] by an even faster and unbiased sampling method called Adaptive DGBS for the valuation of lookback, swing, and barrier option under the VG model through dynamical reset conditions and a preset prior bound of the estimation bias.

In the paper, we will focus on evaluating the lookback and barrier option under the VG model within the Adaptive DGBS framework through both the Monte-Carlo (MC) and randomized Quasi-Monte-Carlo (rQMC) experiments. Specifically, the breakdown of the

paper will be as follows: Chapter 2 reviews the general concept of the Variance-Gamma (VG) process and the DGBS algorithm, proposed by Avramidis et al(2004) [3]. Chapter 3 introduces and discusses details of the Adaptive DGBS algorithm for the VG processes, proposed by Becker (2010)[4]. Chapter 4 describes how one can adapt the Adaptive DGBS in pricing the fair price of the lookback and barrier options with underlying prices under the Variance Gamma processes. Chapter 5 replaces the MC experiment of the Adaptive DGBS pricing framework with the rQMC simulation experiment using a low discrepancy Sobol set in base 2 with a Cranley-Patterson rotation shift. Chapter 6 implements all the algorithms in Chapters 3 through 5 and discusses the Adaptive DGBS method's performance in pricing various options under both MC and rQMC simulations with sensitivity analysis in a floating strike lookback call option. A conclusion and a further improvement suggestion are given in Chapter 7.

Chapter 2

Preliminaries

In this chapter, we will first provide some background for the Variance Gamma (VG) process. Under the motivation of the risk-neutral pricing of the option, we explain how to extend the original VG model with an additional drift term. When simulating the extended variance gamma process for the underlying financial asset, we implement a difference-of-gamma bridge sampling (DGBS) scheme. In Chapter 3, the original DGBS algorithm is refined to a newly developed Adaptive DGBS algorithm proposed by Becker (2010) [4] with fast and exact simulations for final, minimal, and maximal values of the VG process on pricing the path-dependent options like lookback option and barrier option.

2.1 The Variance Gamma (VG) Process

Before the discussions of the Variance Gamma (VG) process, we will provide a general review of the basic definitions on the Wiener and gamma process, where each of them is a Lévy process, a process that has independent and stationary increments according to Bertoin (1998) [5]. In particular, according to Sato (1999)[16], a Lévy process is defined as Definition 2.1.1 below

Definition 2.1.1 (Lévy process) *A stochastic process $\{X_t\}_{t\geq 0}$ is called a Lévy process if it satisfies the following conditions*

1. $X_0 = 0$ *a.s.*
2. *For any* $0 < s < t < \infty$, $X_t - X_s$ *are independent and* $X_t - X_s \overset{d}{=} X_{t-s}$

3. For any $\epsilon > 0$, $\lim_{h\to 0}(|X_{t+h} - X_t| > \epsilon) = 0$, for all $t \geq 0$

For a standard Brownian motion $\{W_t\}_{t\geq 0}$, also known as the Wiener process, the independent increments of the standard Brownian motion are normally distributed with mean zero and variance equals to the size of the increments, i.e. $W_{t+h} - W_t \sim N(0, h)$ for all $t > 0$ and increment $h \geq 0$, where $N(\mu, \sigma^2)$ refers to the normal distribution with mean μ and variance σ^2. In this case, we consider a Brownian motion with drift $\theta \in \mathbb{R}$ and volatility $\sigma > 0$ such that $W_t^{(\theta,\sigma)} = \theta t + \sigma W_t$, $t \geq 0$.

Comparably, for a single-variate gamma process $\{\gamma_t\}_{t\geq 0}$, the increments of the gamma process have gamma distributions with both mean and variance equal to the size of the increments, i.e. $\gamma_{t+h} - \gamma_t \sim \Gamma(h, 1)$, where $\Gamma(a, b)$ is the gamma distribution with mean ab and variance ab^2 for shape parameter a and scale parameter b, and the probability density function of the gamma distribution $\Gamma(a, b)$ is given by

$$f_{\Gamma(a,b)}(x) = x^{a-1}\frac{e^{-\frac{x}{b}}}{b^a\Gamma(a)},\ x \geq 0.$$

However, unlike the Wiener process, the gamma process is discontinuous, similar to the Poisson process, it is non-decreasing for $x \geq 0$ as above equation stated. Similarly, for the two-parameter gamma process $\{\gamma_t^{(\mu,v)}\}_{t\geq 0}$, the increments are gamma distributed with drift $\mu > 0$ and volatility $v > 0$ multiplied by the size of the increments, i.e. the two-parameter gamma process is a process with $\gamma_0^{(\mu,v)} = 0$ and independent increments $\gamma_{t+h}^{(\mu,v)} - \gamma_t^{(\mu,v)} \sim \Gamma(\frac{h\mu^2}{v}, \frac{v}{\mu})$ for $t \geq 0$ and increment $h \geq 0$. Due to the non-decreasing property of gamma process for $x \geq 0$, the gamma process is suitable for being a random time change.

Thus, a Variance Gamma (VG) process $\{\tilde{X}_t{}^{(\theta,\sigma,v)}\}_{t\geq 0}$ with parameters (θ, σ, v) can be obtained by subjecting the Brownian motion $W_t^{(\theta,\sigma)}$ to a random time change that follows a gamma process $\{\gamma_t^{(1,v)}\}_{t\geq 0}$ with drift $\mu = 1$, i.e.

$$\tilde{X}_t{}^{(\theta,\sigma,v)} = W^{(\theta,\sigma)}_{\gamma_t^{(1,v)}} = \theta\gamma_t^{(1,v)} + \sigma W_{\gamma_t^{(1,v)}},\ t \geq 0 \tag{2.1}$$

where the subordinator $\gamma_t^{(1,v)}$ is independent with the subordinated standard Brownian motion $\{W_t\}_{t\geq 0}$, and the VG process $\tilde{X}_t{}^{(\theta,\sigma,v)}$ is also a Lévy process that consists of a deterministic function of time, i.e. a drift, a scaled continuous Brownian motion, and a pure-jump process.

Alternatively, the Variance Gamma (VG) process can also be determined by the difference of two independent gamma processes. Specifically, the difference of gammas repre-

sentation is given by Madan et al (1998)[13]

$$\tilde{X}_t{}^{(\theta,\sigma,v)} = \gamma_t^+ - \gamma_t^-,\ t \geq 0 \tag{2.2}$$

where $\gamma_t^+ := \gamma\left(t; \mu_p, v_p\right) \sim \Gamma(\mu_p^2 t/v_p, v_p/\mu_p)$ and $\gamma_t^- := \gamma\left(t; \mu_n, v_n\right) \sim \Gamma(\mu_n^2 t/v_n, v_n/\mu_n)$ are independent gamma processes with parameters

$$\mu_p = (\sqrt{\theta^2 + 2\sigma^2/v} + \theta)/2,\ \mu_n = \mu_p - \theta,\ v_p = \mu_p^2 v,\ v_n = \mu_n^2 v$$

Since we intend to price the path-dependent option under the VG model with risk-neutral assumptions, we need to add an additional drift term $\mu \in \mathbb{R}$ to the original Variance Gamma process $\tilde{X}_t{}^{(\theta,\sigma,v)}$ such that the extended Variance Gamma process with difference of gammas representation is

$$X_t^{(\mu,\theta,\sigma,v)} = \mu t + \tilde{X}_t{}^{(\theta,\sigma,v)},\ t \geq 0\ . \tag{2.3}$$

Throughout the rest of the paper, we will only use the extended variance gamma process $X_t^{(\mu,\theta,\sigma,v)}$ in the difference of gammas representation.

2.2 Difference-of-Gammas Bridge Sampling (DGBS) of VG

According to both equation (2.1) & (2,2), the sampling of VG process relies crucially on both the time-changed Brownian motion representation and difference of gammas representation. Due to the simpler form of the difference of gammas representation and its ease on applying fast simulation method to sample process values, we will only focus on sampling the gamma process for generating the VG process. For efficient sampling of the gamma process $\gamma_t^{(\mu,v)}$, according to Avramidis et al (2004) [3], the standard method is through gamma sequential sampling (GSS) on a discrete time partition over the interval $[0, T]$ of length 2^m, with $t_i := i\delta T$, $i = 1, \ldots, 2^m$ and $\delta = 2^{-m}$. Based on the discrete time grid, we draw independent gamma increments such that $\gamma_{t_i}^{(\mu,v)} - \gamma_{t_{i-1}}^{(\mu,v)} \sim \Gamma(\mu^2(t_i - t_{i-1})/v, v/\mu)$ for increment $h = t_i - t_{i-1} > 0$ and sum them up to obtain the gamma process overtime.(see Appendix A.1.1)

An alternative approach for sampling gamma process was proposed by Dufresne et al (1991)[11] and is called the Gamma bridge sampling (GBS), which will be our main focus of the algorithm in generating the gamma process throughout the paper. Let $\gamma_t := \gamma_t^{(\mu,v)}$,

i.e. a gamma process with parameters (μ, v), the GBS method mainly depends on the conditional distribution of the gamma process γ_t given the distribution of γ_{τ_1} and γ_{τ_2} for $0 \leq \tau_1 < t < \tau_2$, which is similar to the classical Brownian Bridge sampling. In particular, the conditional distribution of γ_t is given by $\gamma_t = \gamma_{\tau_1} + (\gamma_{\tau_1} + \gamma_{\tau_2})Y$, where $Y \sim Beta((t-\tau_1)\mu^2/v, (\tau_2 - t)\mu^2/v)$. In this case, all variables Y, γ_{τ_1}, and γ_{τ_2} are independent of each other. The probability density function of the beta distribution $Beta(\alpha, \beta)$ is

$$f_{Beta(\alpha,\beta)}(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 y^{\alpha-1}(1-y)^{\beta-1}dy}, \ 0 < x < 1.$$

Different than the gamma sequential sampling (GSS) method, when sampling the gamma process by gamma bridge sampling, we consider the dyadic partition over the interval $[0, T]$ of length 2^m, with depth $m \in \mathbb{N}$. For $M = 2^m$ and $t_i := i\delta T$, $i = 1, \ldots, 2^m$ and $\delta = 2^{-m}$, we first initialize $\gamma_{t_0=0} = 0$ and produce $\gamma_{t_M=T} \sim \Gamma(t_M\mu^2/v, (t_{M/2} - t_M)v/\mu)$ and a random beta variate $Y \sim Beta(t_M\mu^2/v, (t_{M/2} - t_M)\mu^2/v)$. Consequently, we can obtain $\gamma_{t_{M/2}} = \gamma_{t_0} + (\gamma_{t_M} - \gamma_{t_0})Y$, and so on.

In other words, we use both the given distribution for time positions from 0 to $2^m - 1$: the left (start) time, i.e. τ_1, and the given distribution for time positions from 1 to 2^m: the right (end) time, i.e. τ_2, to approximate the conditional distribution of the time position of the average on the sum of the two-time positions, i.e. t, in an iterative manner. (see Appendix A.1.2. for further details) Under these binary time grids, we can improve the efficiency of the quasi-Monte-Carlo technique and fast simulation methods for symmetric beta random variate Y is also applicable.

Eventually, according to Avramidis et al (2004)[3], through integrating the difference of gammas representation (2.3) with gamma bridge sampling method, the so-called difference-of-gammas sampling (DGBS), we can generate a variance gamma process $X_t^{(\mu,\theta,\sigma,v)}$ on a dyadic partition over $t \in [0, T]$. Under this difference-of-gammas sampling (DGBS) framework, we first generate the sequences of $\Gamma_t^+(\mu_p, v_p)$ and $\Gamma_t^-(\mu_n, v_n)$ over the dyadic partition of length 2^m, with time grid $(t_i)_{i=0}^{2^m}$ such that $t_0 = 0, t_1 = T, t_2 = \frac{T}{2}, t_3 = \frac{T}{4}, t_4 = \frac{3T}{4}, \ldots, t_{2^m} = \frac{(2^m-1)T}{2^m}$, or equally defined as follows:

$$t_0 = 0, t_1 = T, t_i = ((2(i - 2^{\lceil log_2(i)-1 \rceil}) - 1)/2^{\lceil log_2(i) \rceil})T, \text{ where } i \in \{2, \ldots, 2^m\}, \tag{2.4}$$

according to Becker (2010) [4]. Consequently, we can obtain the variance gamma process $X_t^{(\mu,\theta,\sigma,v)}$ by the difference of gamma representation, i.e. $X_t^{(\mu,\theta,\sigma,v)} = \mu t + \tilde{X}_t^{(\theta,\sigma,v)} = \mu t + \gamma_t^+(\mu_p, v_p) - \gamma_t^-(\mu_n, v_n)$, t$\geq$0, overtime. (See Appendix A.2.1) Figure 2.1 from below shows an example of one hundred simulations of the Variance Gamma (VG) process using

the difference of gammas sampling (DGBS) method, where we can clearly observed the Variance Gamma process $X_t^{(\mu,\theta,\sigma,\upsilon)}$ as a pure jump process.

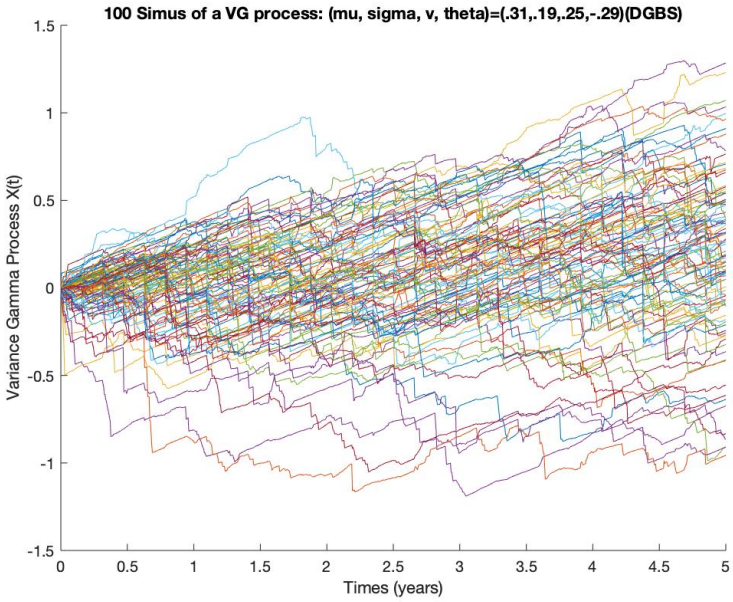


Figure 2.1: Simulations of a variance gamma process using DGBS with Depth m = 10

So far we have introduced and developed the difference of gamma sampling (DGBS) method for simulating the Variance Gamma (VG) process. However, to effectively simulate the underlying financial asset price under the VG model for pricing path-dependent options like the lookback and barrier option, we will introduce an adaptive DGBS algorithm in the next chapter. This algorithm can simulate the final, minimal, and maximal values of the VG process efficiently and provide unbiased valuations of these options.

Chapter 3

Adaptive DGBS

Based on the original DGBS framework described in Chapter 2, Becker (2010) [4] proposed a newly developed algorithm named the adaptive DGBS algorithm that is efficient in generating a lower bound, upper bound, and final value of the VG process. This algorithm, which is discussed further in Chapter 4 and explored numerically in Chapter 6, can prove to be extremely fast and accurate for pricing various path-dependent options under the VG model both empirically and theoretically with a prior error bound.

According to Becker (2010) [4], the original DGBS method gives some useful bounds on the simulated VG process $X_t^{(\mu,\theta,\sigma,v)}$ over the parameter space (μ, θ, σ, v). Consider $X_t = \{X_t\}_{t\geq 0}$ to be any Variance Gamma (VG) process given by (2.3) $X_t = \mu t + \gamma^+ - \gamma^-, t \geq 0$ with drift term $\mu \in \mathbb{R}$, and $\gamma^+ = \{\gamma_t^+\}_{t\geq 0}$ & $\gamma^- = \{\gamma_t^-\}_{t\geq 0}$ independent gamma processes. Due to the non-decreasing property of γ^+ and γ^- explained in Chapter 2., the bounds can be easily developed as follows:

$$\underbrace{\mu\tau_1 + \gamma^+_{\tau_1} - \gamma^-_{\tau_2} + (\tau_2 - \tau_1)\mu^-}_{lb:=L_{[\tau_1,\tau_2]}} \leq X_t \leq \underbrace{\mu\tau_1 + \gamma^+_{\tau_2} - \gamma^-_{\tau_1} + (\tau_2 - \tau_1)\mu^+}_{ub:=U_{[\tau_1,\tau_2]}}, \ \forall t \in [\tau_1, \tau_2] \quad (3.1)$$

where $\mu^- := \min(0, \mu)$, $\mu^+ := \max(0, \mu)$, and obviously, the bounds shrink for small values of $\tau_2 - \tau_1$. (See Corollary 1, Avramidis and L'Ecuyer (2006)[2] for detailed derivations of the bounds.)

3.0.1 Simulating Final and Extremal Values

Inspired by the bounds from (3.1) via sampling the Variance Gamma (VG) process using the DGBS algorithm, an adaptive DGBS algorithm is formulated by Becker (2010)[4] over a refined selection of the original time grid $(t_i)_{i=0}^{2^m}$ to drop the irrelevant simulation steps when possible for pricing typical type of path-dependent options. Consequently, the new time grid becomes $(t_{ji})_{i=0}^{2^m}, j_i < j_{i+1} \forall i$. In particular, the adaptive selection of the time grid for adaptive DGBS will largely depend on the bounds from (3.1).

The bottom line of the adaptive DGBS is to exclude the intervals iteratively from the partition $(t_i)_{i=0}^{2^m}$ that are not giving the minimal or the maximal value of the VG process depending on the types of the option upon one's interests. In other words, for $\mathcal{T}$, the current set of time positions when running the DGBS, if $\min_{t\in\mathcal{T}} X_t \leq L_{[\tau_1,\tau_2]}$ or $U_{[\tau_1,\tau_2]} \leq \max_{t\in\mathcal{T}} X_t$ is satisfied, then the interval $[\tau_1, \tau_2]$ will apparently not contribute to the minimal or the maximal value of the VG process. As a result, the adaptive DGBS scheme will rule out this interval for further iterations.

At the end of the adaptive DGBS scheme for simulating the VG process path, according to Becker (2010) [4], we expect to end up with a comparatively small set $\mathcal{I} = \{I_1, \ldots, I_n\}$, $n \in \mathbb{N}$, of intervals remaining for constructing the maximal and minimal values of the process. In particular, for the remaining intervals $I_j = [\tau_j^l, \tau_j^r]$ for $\mathcal{J} := \{j | j = 1, \ldots, n\}$, these resultant intervals are achieved by using the following useful bounds on the VG process, where

- $sl := \min_{j\in\mathcal{J}} L_{[\tau_j^l,\tau_j^r]}$ is the smallest lower bound among the remaining lower bounds
- $bu := \max_{j\in\mathcal{J}} U_{[\tau_j^l,\tau_j^r]}$ is the biggest upper bound among the remaining upper bounds
- $su_{min} := \min \bigcup_{j\in\mathcal{J}} \{x_{\tau_j^l}, x_{\tau_j^r}\}$ is the smallest upper bound for the minimal x_t
- $bl_{max} := \max \bigcup_{j\in\mathcal{J}} \{x_{\tau_j^l}, x_{\tau_j^r}\}$ is the biggest lower bound for the maximal x_t

Clearly, from the above definition, one can easily deduce that $sl \leq \min_{t\in[0,T]} x_t \leq su_{min}$, and $bl_{max} \leq \max_{t\in[0,T]} x_t \leq bu$, where x_t is the simulated (realized) path of the process X_t. The results are based on the fact that $\min_{t\in[0,T]} x_t$ is the approximated upper bound for the minimum of the process X_t and $\max_{t\in[0,T]} x_t$ is the approximated lower bound for the maximum of the process X_t. In addition, according to Avramidis and L'Ecuyer (2006)[2], the approximations of the minimal, maximal and final values of maximum minimum for the process X_t are $\underline{x} := \frac{1}{2}(sl + su_{min})$ and $\underline{x} := \frac{1}{2}(bu + bl_{max})$, $\bar{X}_T = \max_{t\in[0,T]} x_t$, and $\underline{X}_T = \min_{t\in[0,T]} x_t$ accordingly. Consequently, the approximations' errors of the minimal

Algorithm 1: Simulations of (X_T, $\bar{X}$, $\underline{X}$) from a Variance Gamma Process with Parameter (μ,θ,σ,v) Using Adaptive DGBS by Becker (2010)[4]

input : Variance Gamma process parameters μ, $\theta \in \mathbb{R}$, σ, $v > 0$, terminal time T, depth of the dyadic partition $m \in \mathbb{N}$, prior error bound $\epsilon \geq 0$

output: realization (x, x_l, x_u)of (X_T, $\bar{X}$, $\underline{X}$) from a Variance Gamma Process X_t on $t \in [0, T]$

$\mu_p = (\sqrt{\theta^2 + 2\sigma^2/v} + \theta)/2;\ \mu_n = \mu_p - \theta,\ v_p = \mu_p^2 v,\ v_n = \mu_n^2 v;$

$\mu^+ = \max(0, \mu),\ \mu^- = \min(0, \mu)\ ,\ \gamma_0^+ = 0;\ \gamma_0^- = 0; t = zeros(2^m \times 1);$

$\gamma_1^+ = random\ \Gamma(\mu_p^2 T/v_p, v_p/\mu_p),\ \gamma_1^- = random\ \Gamma(\mu_n^2 T/v_n, v_n/\mu_n);$

$t_0 = 0;\ t_1 = T;\ X = \gamma_1^+ - \gamma_1^- + T\mu;\ bl_{max} = \max(0, X);\ su_{min} = \min(0, X);$

$lint_1^n = 0; rint_1^n = 1; nint^n = 1; pos = 1;$

for $M \leftarrow 1$ **to** m **do**

  $lint^c = lint; rint^c = rint; nint^c = nint; nint = 0;$

  $sl = bl_{max}; bu = su_{min};$

  for $j \leftarrow 1$ **to** $nint^c$ **do**

    $pos = pos + 1;$

    $i_l = lint_j^c;\ i_m = pos;\ i_r = rint_j^c;\ t_{i_m} = \frac{t_{i_r} + t_{i_l}}{2};\ \delta = \frac{t_{i_r} - t_{i_l}}{2v};$

    $Y_p = \text{random } Beta(\delta, \delta);$

    $\gamma_{i_m}^+ = \gamma_{i_l}^+ + Y_p(\gamma_{i_r}^+ - \gamma_{i_l}^+);$

    $Y_n = \text{random } Beta(\delta, \delta);$

    $\gamma_{i_m}^- = \gamma_{i_l}^- + Y_n(\gamma_{i_r}^- - \gamma_{i_l}^-);$

    $x_{i_m} = \gamma_{i_m}^+ - \gamma_{i_m}^-;$

    $su_{min} = min(su_{min}, \gamma_{i_m}^+ - \gamma_{i_m}^- + t_{i_m}\mu);$

    $bl_{max} = max(bl_{max}, \gamma_{i_m}^+ - \gamma_{i_m}^- + t_{i_m}\mu);$

    $i_n = [i_l, i_m]; i_p = [i_m, i_r];$

    for $w \leftarrow 1$ **to** 2 **do**

      $lb = \gamma_{i_n(w)}^+ - \gamma_{i_p(w)}^- + t_{i_n(w)}\mu + (t_{i_p(w)} - t_{i_n(w)})\mu^-;$

      $ub = \gamma_{i_p(w)}^+ - \gamma_{i_n(w)}^- + t_{i_n(w)}\mu + (t_{i_p(w)} - t_{i_n(w)})\mu^+;$

      if $lb \leq su_{min}$ **or** $ub \geq bl_{max}$ **then**

        $nint^n = nint + 1; lint_{nint^n}^n = i_n(w); rint_{nint^n}^n = i_p(w);$

        $sl = \min(sl, lb); bu = \max(bu, ub);$

  if $\max(su_{min} - sl, bu - bl_{max}) \leq 2\epsilon$ **then**

    break

$x_l = \frac{1}{2}(su_{min} + sl), x_u = \frac{1}{2}(bu + bl_{max})$

and maximal values for the process X_t is bounded by $\max(\frac{1}{2}(su_{min} - sl), \frac{1}{2}(bu - bl_{max}))$.

Algorithm 1 from above shows the implementation of the Adaptive DGBS over the dyadic partition, i.e. the time grid $(t_i)_{i=0}^{2^m}$ such that $t_0 = 0, t_1 = T, t_2 = \frac{T}{2}, t_3 = \frac{T}{4}, t_4 = \frac{3T}{4}, \ldots$ of length 2^m with depth m. The algorithm is referenced from Becker (2010)[4] with additional clarifications and it can be easily implemented in any computer software with slight modifications of the indices (line 5). From line 1 to 5, we initialize some known variables such as μ_p and μ^+ etc., then we set $\gamma^+_{t_0=0} = 0$ and generate terminal value of gamma process $\gamma^+_{t_1=T}$ & $\gamma^-_{t_1=T}$ with parameter μ_p, v_p & μ_n, v_n respectively. There are four newly defined variables in line 5, where

- $lint^n$ is the array of the left (start) end position of the interval that might contain either the minimum and maximum of the VG process for the next simulation stage M. It is initialized with position 0 indicating the zero index at time grid $(t_i)_{i=0}^{2^m}$, gamma process, and VG process, i.e. t_0, $\gamma^+_{t_0=0}$, $\gamma^-_{t_0=0}$, and x_0.
- $rint^n$ is the array of the right (end) end position of the interval that might contain either the minimum and maximum of the VG process for the next simulation stage M. It is initialized with number 1 indicating the first index at time grid $(t_i)_{i=0}^{2^m}$, gamma process, and VG process, i.e. t_1, $\gamma^+_{t_1=T}$, $\gamma^-_{t_1=T}$, and x_1.
- $nint^n$ is the array of the number of points within the interval that might contain either the minimum and maximum of the VG process for the next simulation stage M. It is initialized with number 1 indicating there is only one point for the first interval $[t_0, t_1] = [0, T]$
- *pos* is the integer number of the position for evaluation (the mid-point of left and right position) within the interval that might contain either the minimum and maximum of the VG process for the next simulation stage M. It is initialized with number 2 indicating the second index at time grid $(t_i)_{i=0}^{2^m}$, gamma process, and VG process, i.e. t_2, $\gamma^+_{t_2=\frac{t_1+t_0}{2}=\frac{T}{2}}$, $\gamma^-_{t_2=\frac{t_1+t_0}{2}=\frac{T}{2}}$, and x_2, to be evaluated

Similarly, $lint^c$, $rint^c$, and $nint^c$ are the corresponding positions in the current iteration stage M. Specifically, these variables for recording the course of the positions are all arrays with the length depending on the prior error bound that will be introduced later in this chapter. Each entry is in accordance with each simulated path position. Thus, through the iteration stage over the depth m from line 6 to line 25, we can dynamically subdivide the interval according to the dyadic partitions and store the indispensable interval in each simulation path. Eventually, we can approximate the minimal, maximal and final values

of the VG process using the derived formula in line 28. According to Becker (2010) [4], line 26 to 27, i.e. the breaking condition, is based on the proposition as follows:

Proposition 3.0.1 *Let the prior error bound $\epsilon \geq 0$. If the depth m is set to ∞ and the breaking condition:*

$$\textit{if } \max(\frac{1}{2}(su_{min} - sl), \frac{1}{2}(bu - bl_{max})) < \epsilon \textit{ then break} \tag{3.2}$$

is inserted at line 27 to 28, then the error bounds:

$$|\underline{x} - \min_{t\in[0,T]}| < \epsilon \textit{ and } |\bar{x} - \max_{t\in[0,T]}| < \epsilon$$

hold after termination of the algorithm.

It is also worthy to note that lines 20 to 25 are based on the bounds determined in (3.1) to investigate the current interval of interest and whether or not it contributes to the minimal or maximal of X_t. Furthermore, for each simulated path, we need to update sl and bu for the maximum bl_{max} and the minimum su_{min} accordingly in line 8. In line 3, the random gamma variates are generated via inversion using the inverse gamma function with a random uniform variate over the domain [0,1].

To generate the symmetric beta distribution $B(a,a)$ in line 12 and 14, for $a > \frac{1}{2}$, we use the fast one-liner method presented by Devroye (1996) [10], i.e.

$$B(a,a) \overset{\mathcal{L}}{=} \frac{1}{2} + \frac{S}{2\sqrt{1 + \frac{1}{U^{-\frac{1}{a}}\cos(2\pi V)^2}}}, \text{ for } a > \frac{1}{2} \tag{3.3}$$

where S is a random sign defined by $S = \mathbb{1}_{\{V_1 \leq 1/2\}}$ for V_1 a random uniform over [0,1], and U and V are other independent random uniforms over [0,1].

On the other hand, for $0 < a \leq \frac{1}{2}$, we implement the Ulrich's polar method for symmetric beta random variates by Devroye (1986) [9], i.e. when U, V are independent random uniforms over [0,1] and [-1,1] accordingly, we repeat the generations step until $S = U^2 + V^2 \leq 1$, then it returns the symmetric beta $B(a,a)$ as the following

$$B(a,a) \overset{\mathcal{L}}{=} \frac{1}{2} + \frac{UV}{S}\sqrt{1 - S^{\frac{2}{2a-1}}}, \text{ for } 0 < a \leq \frac{1}{2}. \tag{3.4}$$

Now that we have introduced and developed the main algorithm of the research, we can move forward to derive the risk-neutral scheme for pricing the lookback, swing, and

barrier options under the VG model using this newly developed adaptive DGBS algorithm in the next chapter.

Chapter 4

Applications to Option Pricing

Overall, there is no unique martingale measure for a Lévy process as the process is a jump process, from which it is also the case for the Variance Gamma (VG) process. Thanks to the works by Madan et al (1998)[13], they developed a risk-neutral formula to resolve this problem for options with underlying financial asset prices in a exponential form of Variance Gamma (VG) process. In particular, let $\{S_t\}_{t\geq 0}$ be the risk-neutral price of an underlying financial asset on a probability space $(\Omega, (\mathcal{F}_t)_{t\geq 0}, \mathbb{Q})$. Under the arbitrage-free assumption and the risk-neutral measure $\mathbb{Q}$, the asset price $\{S_t\}_{t\geq 0}$ dynamics for a variance gamma process $\{\tilde{X}_t{}^{(\theta,\sigma,v)}\}$ with constant continuous compounding dividend yield q and constant risk-free interest rate r are given by

$$S_t = S_0 \exp\underbrace{\big(\underbrace{(\omega + r - q)}_{\mu} t + \tilde{X}_t{}^{(\theta,\sigma,v)}\big)}_{X_t^{(\mu,\theta,\sigma,v)}},\ t \geq 0 \tag{4.1}$$

where we add an additional drift term $\mu = (\omega + r - q) \in \mathbb{R}$ to the process $\tilde{X}_t{}^{(\theta,\sigma,v)}$ that gives us a process $X_t^{(\mu,\theta,\sigma,v)} = \mu t + \tilde{X}_t{}^{(\theta,\sigma,v)},\ t \geq 0$, and the constant $\omega = \ln(1 - \theta v - \sigma^2 v/2)/v$ is such that the discounted asset price is a martingale, i.e.

$$\mathrm{E}[e^{-(r-q)t} S_t] = S_0. \tag{4.2}$$

In this case, we require $1 - \theta v - \sigma^2 v/2 > 0 \Rightarrow v(\theta - \sigma^2/2) < 1$ to make sure $\mathrm{E}[S_t]$ is finite for all $t > 0$. Throughout the rest of the paper, we will assume $v(\theta - \sigma^2/2) < 1$ without further specifications for risk-neutral pricing.

In this paper, we consider pricing the path-dependent options under the VG model.

According to Avramidis and L'Ecuyer (2006)[2], for finite number of observation times $0 = t_0 < t_1 < \ldots < t_n = T$, the value of the path-dependent option at maturity T is given by $c = \mathrm{E}[C]$ with the random variable C defined as

$$C = e^{-rT} f(S_{t_1}, \ldots, S_{t_n})$$

with the payoff function $f : \mathbb{R}^n \to \mathbb{R}$. On the other hand, in the continuous case, for $t \in [0, T]$ with equal-length subinterval, the value of the discounted payoff random variable C can be expressed as

$$C = e^{-rT} f(\{S_t\}_{0 \le t \le T}).$$

In our case, the adaptive DGBS algorithm renders us a concept of Monte-Carlo valuation approach for options that only depends on final, minimal or maximal values of the underlying asset prices. In other words, given N simulations of $(x_k, \underline{x}_k, \bar{x}_k)$ at time $T > 0$ for each $k \in \{1, \ldots, N\}$ of the VG process, we can obtain the corresponding realizations of the risk-neutral price from (4.1) for the underlying asset as $(s_k, \underline{s}_k, \bar{s}_k)$ at time $T > 0$. The estimated fair price of the options written upon the underlying asset is given by

$$\hat{C} = e^{-rT} \times \frac{1}{N} \sum_{k=1}^{N} f(s_k, \underline{s}_k, \bar{s}_k) \tag{4.3}$$

with designate payoff function f chosen upon different types of options. A simple implementation of this option pricing framework is the valuation of the swing option with $f(s_k, \underline{s}_k, \bar{s}_k) = \bar{s}_k - \underline{s}_k$. The standard error of the corresponding estimator, i.e. $\hat{C}$, for the fair price of the option is

$$se = \sqrt{\frac{\sum_{k=1}^{N} (f(s_k, \underline{s}_k, \bar{s}_k) - \hat{C})^2}{N(N-1)}} \tag{4.4}$$

4.1 Lookback Options

In this section, we will adjust the adaptive DGBS to Algorithm 2 to determine the fair price of a floating strike lookback call option under the VG model in a dynamic setting based on a dyadic time partitions with depth m. Specifically, a floating strike lookback call option, different than the fixed strike lookback option for the fixed strike price at purchase, has the strike at the lowest price of the underlying asset over the option's lifespan.

Algorithm 2: Valuation of Floating-Strike Call Option under the VG model with Parameter (μ,θ,σ,v) Using Adaptive DGBS by Becker (2010)[4]

input : Variance Gamma process parameters μ, $\theta \in \mathbb{R}$, σ, $v > 0$, terminal time T, spot price of the underlying S_0, risk-free interest rate r, constant dividend yield q, number of simulations run $N \in \mathbb{N}$, error bound $\epsilon \geq 0$, (maximum) depth of the dyadic partition $m \in \mathbb{N}$

output: Estimated fair price C with standard error se of floating strike call

$\mu_p = (\sqrt{\theta^2 + 2\sigma^2/v} + \theta)/2;\ \mu_n = \mu_p - \theta,\ v_p = \mu_p^2 v,\ v_n = \mu_n^2 v;$
$\mu^+ = \max(0, \mu),\ \mu^- = \min(0, \mu)\ ,\ \gamma_0^+ = 0;\ \gamma_0^- = 0;$
$t = zeros(2^m \times 1); t_0 = 0;\ t_1 = T;$
for $k \leftarrow 1$ **to** N **do**
  $\gamma_1^+ = random\ \Gamma(\mu_p^2 T/v_p, v_p/\mu_p),\ \gamma_1^- = random\ \Gamma(\mu_n^2 T/v_n, v_n/\mu_n);$
  $X = \gamma_1^+ - \gamma_1^- + T\mu; bl_{max} = \max(0, X); su_{min} = \min(0, X);$
  $lint_1^n = 0; rint_1^n = 1; nint^n = 1; pos = 1;$
  for $M \leftarrow 1$ **to** m **do**
    $lint^c = lint; rint^c = rint; nint^c = nint; nint = 0; sl = bl_{max};$
    for $j \leftarrow 1$ **to** $nint^c$ **do**
      $pos = pos + 1;$
      $i_l = lint_j^c;\ i_m = pos;\ i_r = rint_j^c;\ t_{i_m} = \frac{t_{i_r} + t_{i_l}}{2};\ \delta = \frac{t_{i_r} - t_{i_l}}{2v};$
      $Y_p = \text{random } Beta(\delta, \delta);$
      $\gamma_{i_m}^+ = \gamma_{i_l}^+ + Y_p(\gamma_{i_r}^+ - \gamma_{i_l}^+);$
      $Y_n = \text{random } Beta(\delta, \delta);$
      $\gamma_{i_m}^- = \gamma_{i_l}^- + Y_n(\gamma_{i_r}^- - \gamma_{i_l}^-);$
      $su_{min} = min(su_{min}, \gamma_{i_m}^+ - \gamma_{i_m}^- + t_{i_m}\mu);$
      $i_n = [i_l, i_m]; i_p = [i_m, i_r];$
      for $w \leftarrow 1$ **to** 2 **do**
        $lb = \gamma_{i_n(w)}^+ - \gamma_{i_p(w)}^- + t_{i_n(w)}\mu + (t_{i_p(w)} - t_{i_n(w)})\mu^-;$
        $ub = \gamma_{i_p(w)}^+ - \gamma_{i_n(w)}^- + t_{i_n(w)}\mu + (t_{i_p(w)} - t_{i_n(w)})\mu^+;$
        if $lb \leq su_{min}$ **then**
          $nint^n = nint + 1; lint_{nint^n}^n = i_n(w); rint_{nint^n}^n = i_p(w);$
          $sl = \min(sl, lb);$
    if $e^{-rT} S_0(e^{su_{min}} - e^{sl})) \leq 2\epsilon$ **then**
      break
  Payoff(k) $= e^{-rT} S_0(e^X - \frac{1}{2}(e^{su_{min}} + e^{sl}));$
$C = \frac{1}{N} sum(Payoff); se = \sqrt{var(Payoff)/N};$

In this case, the discounted payoff of the floating strike lookback call option conditioned on the simulated $(s_t)_{t\in[0,T]}$ of the underlying asset price process on $[0,T]$ is

$$\text{Payoff}_{original} = e^{-rT}\max(s_T - \inf_{t\in[0,T]} s_t, 0) = e^{-rT}[s_T - \inf_{t\in[0,T]} s_t] \tag{4.5}$$

According to Avramidis and L'Ecuyer (2006)[2], they compare the estimated bias of different estimators involving the Richardson's extrapolation to the limits. Consequently, Avramidis and L'Ecuyer find a best performing class of estimator for the fair price of the floating strike lookback call option without extrapolation as the following estimated discounted payoff function

$$\begin{aligned}\text{Payoff}_{best} &= \frac{1}{2}[e^{-rT}(S_0e^x - S_0e^{su_{min}}) + e^{-rT}(S_0e^x - S_0e^{sl})] \\ &= e^{-rT}S_0[e^x - \frac{1}{2}(e^{su_{min}} + e^{sl})]\end{aligned} \tag{4.6}$$

where $x = \gamma_1^+ + \gamma_1^- + T\mu$ is the terminal VG process over the dyadic partitions. Clearly, the approximation error, i.e. the difference between the original payoff function (4.5) and the best estimator by Avramidis and L'Ecuyer (4.6), is bounded by $e^{-rT}S_0(e^{su_{min}} - e^{sl})/2$. The bound is obtained by the fact that $s_T = S_0e^x$ and $S_0e^{sl} \leq \inf_{t\in[0,T]} s_t \leq S_0e^{su_{min}}$.

Thus, based on (4.6) and the approximation error's bound, the finalized discounted payoff of the floating strike lookback call option is (indicated in line 28 from Algorithm 2)

$$\hat{C}_{floating} = e^{-rT} \times \frac{1}{N}\sum_{k=1}^{N}(S_0[e^{x_k} - \frac{1}{2}(e^{su_{min_k}} + e^{sl_k})]) \tag{4.7}$$

with breaking condition determined by the approximation error's bound, i.e.

$$e^{-rT}S_0(e^{su_{min}} - e^{sl})/2 \leq \epsilon$$

for a prior error bound $\epsilon \geq 0$ similarly defined as Proposition 3.1.1. The breaking condition is indicated in the outer part of each M stage iteration in line 25 to 26 from Algorithm 2. The standard error of the estimator for the floating strike lookback call option is (indicated in line 28)

$$se_{\text{floating}} = \sqrt{\frac{\sum_{k=1}^{N}(\text{Payoff}_{\text{ best}}(k) - \hat{C}_{\text{floating}})}{N(N-1)}}. \tag{4.8}$$

In light of what is adjusted in Algorithm 2 comparing to Algorithm 1, since we are pricing the floating strike lookback option, we only need the intervals that contribute to the minimal value of the process path at each stage M to determine the corresponding payoff. Consequently, in line 17, 22, and 24, we only include sl and su_{min}, i.e. the smallest of the remaining lower bounds and the smallest upper bound for the minimum accordingly, to determine the payoff of the option at each simulation $k \in \{1, \ldots, N\}$. However, we still need bl_{max}, i.e. the largest lower bound for the maximum, to initialize the value of sl at each M stage iteration in line 9. Moreover, function $var(\cdot)$ in line 28 is referring to the sample variance formula and function $sum(\cdot)$ is the arithmetic sum formula. Other parts of Algorithm 2 for estimating the fair price of the floating strike lookback option with underlying price process under the VG model are the same as Algorithm 1 In particular, the random gamma variate is generated by inversion, and the symmetric beta random variates in lines 13 and 15 are generated through fast simulation methods (3.3) and (3.4) upon different values of the shape parameters.

4.2 Barrier Options

The adaptive DGBS algorithm in Algorithm 1 can also be adjusted to accommodate the estimation for the fair price of the barrier call option (up-and-in calls and down-and-out calls) over the same dyadic partition of length 2^m as in Algorithm 1 and Algorithm 2. Specifically, an up-and-in/down-and-out call, or the so-called knock-in/knock-out call option, is defined as follows. For an upper barrier $B_u > S_0$, if the underlying price reached the upper barrier at any time, the up-and-in call option is activated, and the holder of the option receives the payoff at maturity T, or otherwise, the option becomes worthless at expiry. Besides, the payoff of the up-and-in option at time T is exactly the payoff of a European vanilla call option, i.e. $\max(s_T - K)$ with fixed strike K, for simulated s_T. A down-and-out option has the opposite characteristic as the up-and-in option. For a lower barrier $B_l < S_0$, as long as the lower barrier B_l is not reached, the holder of the down-and-out call option has the same payoff of a plain vanilla call at maturity T.

Thus, the discounted payoff of a up-and-in barrier call option conditioned on the simulated $(s_t)_{t\in[0,T]}$ of the underlying asset price process on $[0, T]$ in a dynamic setting is

$$\text{Payoff}_{up-in} = e^{-rT} \max(0, s_T - K) \mathbb{1}\{\sup_{t\in[0,T]} s_t > B_u\} \tag{4.9}$$

where B_u is the upper barrier of the option for $S_0 < B_u$ and $\mathbb{1}\{\cdot\}$ is the indicator function. Similarly, the down-and-out barrier call option conditioning on the simulated $(s_t)_{t\in[0,T]}$ of

Algorithm 3: Valuation of Barrier (up-in and down-out) Call Option under the VG model with Parameter (μ,θ,σ,v) Using Adaptive DGBS by Becker (2010)[4]

$\mu_p = (\sqrt{\theta^2 + 2\sigma^2/v} + \theta)/2;\ \mu_n = \mu_p - \theta,\ v_p = \mu_p^2 v,\ v_n = \mu_n^2 v;$
$\mu^+ = \max(0, \mu),\ \mu^- = \min(0, \mu)\ ,\ \gamma_0^+ = 0;\ \gamma_0^- = 0;$
$t = zeros(2^m \times 1); t_0 = 0;\ t_1 = T;$
for $k \leftarrow 1$ **to** N **do**
  $\gamma_1^+ = random\ \Gamma(\mu_p^2 T/v_p, v_p/\mu_p),\ \gamma_1^- = random\ \Gamma(\mu_n^2 T/v_n, v_n/\mu_n);$
  $X = \gamma_1^+ - \gamma_1^- + T\mu; bl_{max} = \max(0, X); su_{min} = \min(0, X);$
  $lint_1^n = 0; rint_1^n = 1; nint^n = 1; pos = 1;$
  $\#in = false$; #for down-out option;
  if *conditions met for* Θ **then**
    $in = true$; $done = false$;
    for $M \leftarrow 1$ **to** m **do**
      $lint^c = lint; rint^c = rint; nint^c = nint; nint = 0; sl = bl_{max};$
      for $j \leftarrow 1$ **to** $nint^c$ **do**
        $pos = pos + 1;$
        $i_l = lint_j^c;\ i_m = pos;\ i_r = rint_j^c;\ t_{i_m} = \frac{t_{i_r} + t_{i_l}}{2};\ \delta = \frac{t_{i_r} - t_{i_l}}{2v};$
        $Y_p = \text{random } Beta(\delta, \delta);$
        $\gamma_{i_m}^+ = \gamma_{i_l}^+ + Y_p(\gamma_{i_r}^+ - \gamma_{i_l}^+);$
        $Y_n = \text{random } Beta(\delta, \delta);$
        $\gamma_{i_m}^- = \gamma_{i_l}^- + Y_n(\gamma_{i_r}^- - \gamma_{i_l}^-);$
        $su_{min} = min(su_{min}, \gamma_{i_m}^+ - \gamma_{i_m}^- + t_{i_m}\mu);$
        $bl_{max} = max(bl_{max}, \gamma_{i_m}^+ - \gamma_{i_m}^- + t_{i_m}\mu);$
        $i_n = [i_l, i_m]; i_p = [i_m, i_r];$
        for $w \leftarrow 1$ **to** 2 **do**
          $lb = \gamma_{i_n(w)}^+ - \gamma_{i_p(w)}^- + t_{i_n(w)}\mu + (t_{i_p(w)} - t_{i_n(w)})\mu^-;$
          $ub = \gamma_{i_p(w)}^+ - \gamma_{i_n(w)}^- + t_{i_n(w)}\mu + (t_{i_p(w)} - t_{i_n(w)})\mu^+;$
          if *knock-out or knock-in condition met for* Θ **then**
            $in = true; done = true;$ #for up-in option
            $\#in = false; done = true;$ #for down-out option
          if *current interval is not yet definite for* Θ **then**
            $nint^n = nint + 1; lint_{nint^n}^n = i_n(w); rint_{nint^n}^n = i_p(w);$
      if *done==true* ***or*** *(*$nint^n == 0$*)* **then break**;
  if *knock-in/knock-out condition does not/does meet for* Θ **then** $in = false$;
  if $in == true$ **then**
    Payoff$(k) = e^{-rT} max(0, S_0 e^X - K)$
  else Payoff$(k) = 0$;
$C = sum(Payoff)/N; se = \sqrt{var(Payoff)/N}$

the underlying asset price process on $[0, T]$ has its payoff function as

$$\text{Payoff}_{down-out} = e^{-rT} \max(0, s_T - K) \mathbb{1}\{\inf_{t\in[0,T]} s_t > B_l\} \tag{4.10}$$

where $B_l < S_0$ is the lower barrier and K is the fixed strike price. According to Becker (2010) [4], in arbitrary depth M, for $M \in \{1, \ldots, m\}$, of the dyadic partition for the difference of gammas representation of the underlying price $s_t = e^{\mu t + \gamma_t^+ - \gamma_t^-}$, $t \in [0, T]$, there exists expressions for the lower and upper bounds for the up-and-in call's payoff over time, where

$$\text{Payoff}_{up-Lower} = e^{-rT} \max(0, s_T - K) \mathbb{1}\{S_0 e^{bl_{max}} > B_u\} \tag{4.11}$$

and,

$$\text{Payoff}_{up-Upper} = e^{-rT} \max(0, s_T - K) \mathbb{1}\{S_0 e^{ub} > B_u\} \tag{4.12}$$

in a adaptive DGBS framework such that $\text{Payoff}_{up-Lower} \leq \text{Payoff}_{up-in} \leq \text{Payoff}_{up-Upper}$. The right inequality is due to the fact that the biggest lower bound for the maximum price is smaller or equal to the maximal price of the underlying. The left inequality is because the biggest upper bound of the remaining upper bound of the underlying price is larger or equal than the the maximal price of the underlying. The down-and-out call option's payoff's estimated upper and lower bound can be derived in a similar manner with su_{min} and sl.[1][2][3] These estimators, i.e. (4.9), (4.11), and (4.12), converge to each other if at least one of the following conditions is met

- $S_0 e^x - K = S_0 e^{\gamma_1^+ - \gamma_1^- + T\mu} - K \leq 0$ for zero payoff of a call option
- $S_0 e^{bl_{max}} \leq S_0 e^{ub} \leq B_u$ for indicator function equals to zero
- $S_0 e^{ub} \geq S_0 e^{bl_{max}} \geq B_u$ for indicator function equals to one

where we can clearly observe that, in contrast with the payoff of a lookback option that solely on the extremal values over continuous time, the payoff of the barrier option depends on the extremal values only at particular points where it hits the barrier. Thus, according

[1] The estimated lower bound for the down-and-out calls' payoff is $\text{Payoff}_{down-Lower} = e^{-rT} \max(0, s_T - K) \mathbb{1}\{S_0 e^{sl} > B_l\}$

[2] The estimated upper bound for the down-and-out calls' payoff is $\text{Payoff}_{down-Upper} = e^{-rT} \max(0, s_T - K) \mathbb{1}\{S_0 e^{su_{min}} > B_l\}$

[3] In a adaptive DGBS framework, $\text{Payoff}_{down-Lower} \leq \text{Payoff}_{down-out} \leq \text{Payoff}_{Down-Upper}$ holds

to Becker (2010)[4], if the maximum depth M is set to arbitrary large integer, the payoff bounds from above will match at the end of the iterations, where the prior bound ϵ for the approximation error will be exactly equal to zero and thereby we will not need to initialize a prior error bound ϵ in this case.

Therefore, in light of this concept, we can adjust the Adaptive DGBS to evaluate the fair price of the barrier option, specifically the up-and-in and down-and-out call options, under the VG model in dynamic reset conditions as shown in Algorithm 3.

We first initialize everything from line 1 to 8. In particular, in line 8, for an down-and-out call option, we set $in = false$ to make sure if the conditions in line 9 are not satisfied meaning that the down-and-out call option is worthless in this case, we can assign the zero payoff by lines 31 to 34. Later, by imposing the necessary conditions in line 9, we ensure the positive payoff using the first lower bound $S_0 e^{-\gamma_1^- - T\min(0,\mu)}$, and the first upper bound $S_0 e^{\gamma_1^+ + T\max(0,\mu)}$ for the price process in accordance to different types of barrier option before the M stage iteration. After fulfilling the positive payoff condition in line 9, we start the adaptive DGBS algorithm with slight adjustments, where at each stage M, the knock-in and knock-out conditions are inspected. Specifically, from line 26 to 27, for an up-in call in knock-in condition, once the upper barrier is crossed, we stop the current iteration by setting *done* to *true* and set the Boolean indicator *in* to *true* indicating the up-in option is activated, and its payoff is given by line 33 as the plain vanilla call option's payoff. However, if the bound is not yet definite according to line 28, we continue the search as line 29 stated by checking if $S_0 e^{lb} < B_l$ for a down-and-out call or $S_0 e^{ub} > B_u$ for an up-and-in call. A similar approach can be applied to a down-and-out call option with specified knock-out conditions in both lines 9 and 26. Furthermore, the beta and gamma random variates are simulated in the same way as we described in the floating strike lookback option case. The estimated price and standard error of an up-and-in call option using Adaptive DGBS are in line 35.

Table 4.1 from below summarizes the typical conditions for different types of knock-in and knock-out barrier options indicating as the parameter vector Θ in Algorithm 3. For more detailed information regarding the other types of knock-in or knock-out options, one can refer to the paper by Becker (2010)[4].

Line	Condition Type	Option Type	Condition/value
9	Initial Conditions	up-and-in call option	$S_0e^x \geq K$ $S_0e^{\gamma_1^+ + T\max(0,\mu)} > B_u$
		down-and-out call option	$S_0e^x \geq K$ $S_0e^{-\gamma_1^- - T\min(0,\mu)} > B_l$
26, 31	Knock-in or Knock out conditions	up-and-in call option (knock-in)	$S_0e^{bl_{max}} > B_u$
		down-and-out call option (knock-out)	$S_0e^{su_{min}} < B_l$ and set $in = false$
28	Relevance conditions	up-and-in call option	$S_0e^{ub} > B_u$
		down-and-out call option	$S_0e^{lb} < B_l$

Table 4.1: Conditions/payoffs of the Adaptive DGBS on Valuation of Barrier Option

So far we have investigated how adaptive DGBS algorithm can be adjusted and applied to price various types of path-dependent options under the VG model in a dynamic setting using Monte-Carlo (MC) simulations. Due to the special binary partition over the time grid, it is also interesting to see if we can apply the randomized Quasi-Monte-Carlo method to evaluate those options under the VG model through the adaptive DGBS algorithm to reduce the variance of estimations. Thus, as part of my own research, in the next chapter, we will discuss the application of rQMC on Algorithms 2 and 3 using a low-discrepancy Sobol sequence with a simple randomized Cranley-Patterson Rotation shift to reduce the estimation's variance.

Chapter 5

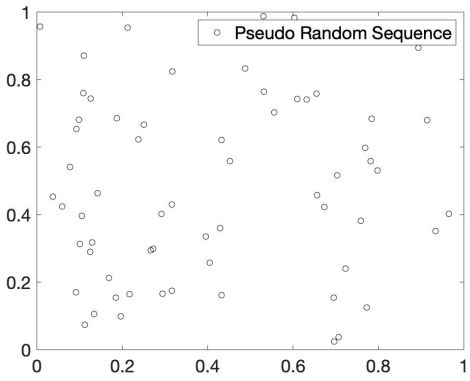
Randomized Quasi-Monte-Carlo

Due to the dyadic partition over the adaptive DGBS framework, a randomized Quasi-Monte-Carlo (rQMC) method can be applied to the original algorithm in pricing the look-back option and barrier option using a low-discrepancy Sobol sequence with a Cranley-Patterson Rotation shift. The basic idea of the rQMC method is to replace the random sampling method used in Monte-Carlo simulation as in Algorithms 1, 2, and 3, i.e. the random uniform variate, by a sampling scheme obtained by taking a low-discrepancy point set P_n and adding a randomized shift. It can be showed that the resulting quasirandom sequence is more uniform than pesudorandom sequence as demonstrated in Figure 5.1. Consequently, due to the lower discrepancy, a smaller variance can be obtained through using the rQMC method, especially in a dyadic (or binary) time grid in our case with a low-discrepancy point set in base 2 like the Sobol sequence. Specifically, the formal definition of the low-discrepancy sequence is defined as follows according to Lemieux (2009) [12]

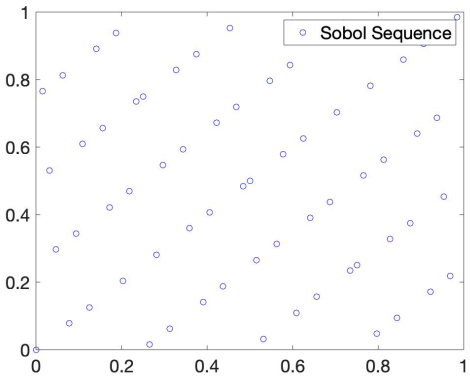
Definition 5.0.1 *Low-discrepancy sequence (Lemieux (2009)[12])*
Consider a sequence with s-dimensional points $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots$ *over* $[0,1)^s$. *Let* P_n *denote the point set containing the first n points, i.e.* $P_n = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n\}$. *For vector* $\boldsymbol{v} = (v_1, \ldots, v_s) \in [0,1)^s$, *denote* $\alpha(P_n, \boldsymbol{v})$ *as the cardinality of the set* $\{\boldsymbol{u}_i : 0 \leq u_{i,j} \leq v_j, i = 1, \ldots, n\}$, *i.e.* $\alpha(P_n, v)$ *refers to how many points* u_i *falls in the box* $B(v) = \{\boldsymbol{u} \in [0,1)^s : 0 \leq u_j \leq v_j, 1 \leq j \leq s\}$ *at the end. The star discrepancy (or measure of how the point set deviate from the uniform distribution through concept of discrepancy) of* P_n *is denoted as* $D^*(P_n) = \sup_{\boldsymbol{v} \in [0,1)^s} |v_1 \cdots v_s - \alpha(P_n, \boldsymbol{v})/n|$. *The sequence of* $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots$ *is a low-discrepancy sequence if for a finite point set* P_n

$$D^*(P_n) \in \mathcal{O}(n^{-1}(log(n))^s)$$

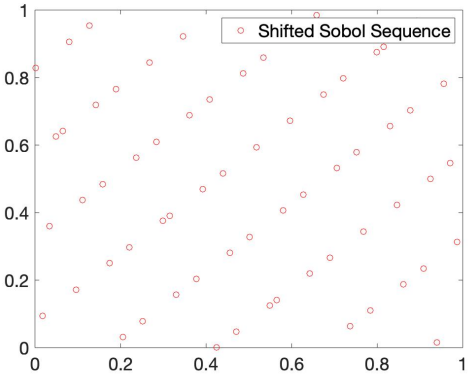
According to the Definition 5.0.1, the Sobol sequence is one of the low-discrepancy sequences using a recurrence in base 2. Based on Figure 5.1, the (x,y) coordinate generated by the Sobol sequence in (a) is more uniform by the one generated by the pseudorandom sequence (i.e. the random uniform) in (b). Furthermore, the Sobol sequence



(a) Pseudorandom Sequence (runiform)



(b) Sobol Sequence

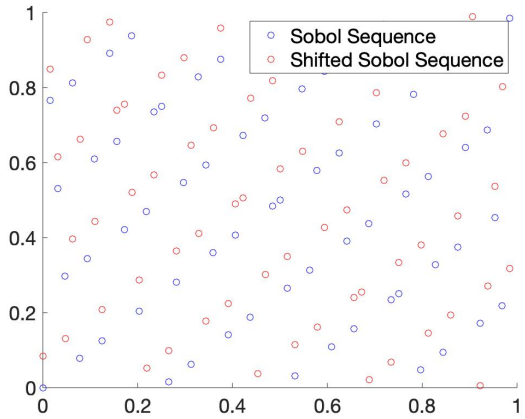


(c) Shifted Sobol Sequence

Figure 5.1: Three Different Point Sets with 2^6 Points

after a randomization shift, i.e. the Cranley-Patterson Rotation, preserved its uniformity as (c) indicated. In particular, according to Lemieux (2009) [12], the Cranley-Patterson Rotation is achieved by shifting the original point $u_i \in [0,1)^s$ into $\tilde{u_i} = (u_i + r)\ mod\ 1$ for $r \in \mathcal{U}([0,1)^s)$ illustrated in Figure 5.2. In this case, $\mathcal{U}([0,1)^s)$ refers to the s-dimensional uniform distribution over the domain $[0,1)^s$.

Figure 5.2: Illustration of Cranley-Patterson Rotation on Sobol Sequence with 2^6 Points



To adjust the Adaptive DGBS for the implementation of the rQMC on pricing the lookback and barrier option with the underlying price under the VG model, before the inner Monte-Carlo simulation for each i, $i \in \{1, \ldots, N_{rQMC}\}$, we first consider creating a Sobol point set $P_{N_{rQMC}} = \{u_1, \ldots, u_{N_{rQMC}}\}$ with recurrence in base 2. Subsequently, we generate random uniform vectors $\mathbf{r} \sim \mathcal{U}([0,1)^s)$ and apply the Cranley-Patterson Rotation for each point u_i in $P_{N_{rQMC}}$, $i \in \{1, \ldots, N_{rQMC}\}$, such that

$$\tilde{u_i} = (u_i + \mathbf{r}) \; mod \; 1 \sim \mathcal{U}([0,1)^s) \text{ for any } u_i \tag{5.1}$$

where we defined the shifted Sobol point set $\tilde{P}_{N_{rQMC}}(\mathbf{r}) = \{\tilde{u_i} | (u_i + \mathbf{r}) \; mod \; 1, 1 \leq i \leq N_{rQMC}\}$. For independent vectors $\mathbf{r}_1, \ldots, \mathbf{r}_n$, each $\mathbf{r}_k \in \mathcal{U}([0,1)^s)$, the unbiased estimator of the fair price for the option over the dyadic partition with depth m and length 2^m is given by

$$\hat{C}_{rQMC,r} = \frac{1}{n} \sum_{k=1}^{n} \left(e^{-rT} \frac{1}{N_{rQMC}} \sum_{i=1}^{N_{rQMC}} f(s_i(\tilde{u}_i^{r_k}), \underline{s_i}(\tilde{u}_i^{r_k}), \bar{s}_i(\tilde{u}_i^{r_k})) \right) \tag{5.2}$$

where f is the payoff function of the option and n is the number of copies for the shifted Sobel set. Each $\tilde{u}_i^{r_k}$ with shift r_k and s coordinates is applied to generate the final and extremal realizations of the underlying asset price $(s_i, \underline{s}_i, \bar{s}_i)$ at time T. The value of the dimension s is depending on the maximum depth m of the path in each simulation i, where each quasirandom number $\tilde{u}_{i,j}^{r_k}$ is used to generate one random variate at each path $j \in \{1, \ldots, s\}$.

Algorithm 4: Valuation of Floating-Strike Call Option under the VG model with Parameter (μ,θ,σ,v) with Adaptive DGBS Using rQMC Method

input : Variance Gamma process parameters μ, $\theta \in \mathbb{R}$, σ, $v > 0$, terminal time T, spot price of the underlying S_0, risk-free interest rate r, constant dividend yield q, number of simulations run $N_{rQMC} \in \mathbb{N}$, error bound $\epsilon \geq 0$, (maximum) depth of the dyadic partition $m \in \mathbb{N}$, the numbers of the Sobol sequence generate $n \in \mathbb{N}$

output: Estimated fair price C with standard error se of floating strike call

$\mu_p = (\sqrt{\theta^2 + 2\sigma^2/v} + \theta)/2;\ \mu_n = \mu_p - \theta,\ v_p = \mu_p^2 v,\ v_n = \mu_n^2 v;$

$\mu^+ = \max(0, \mu),\ \mu^- = \min(0, \mu)\ ,\ \gamma_0^+ = 0;\ \gamma_0^- = 0;$

$t = zeros(2^m \times 1); t_0 = 0;\ t_1 = T;$

for $p \leftarrow 1$ **to** n **do**

  $sobol = sobolset(2 \times (2^m + 1)$ by $N_{rQMC});$

  $r = \text{random } \mathcal{U}(0,1);\ \#(2 \times (2^m + 1))$ by 1 Vector;

  $sobol_s = (sobol + r)_{\text{columnwize}} \bmod 1;$

  for $k \leftarrow 1$ **to** N_{rQMC} **do**

    $\gamma_1^+ = \Gamma^{-1}(sobol_s(k,1), \mu_p^2 T/v_p, v_p/\mu_p),\ \gamma_1^- = \Gamma^{-1}(sobol_s(k,2), \mu_n^2 T/v_n, v_n/\mu_n);$

    $X = \gamma_1^+ - \gamma_1^- + T\mu; bl_{max} = \max(0, X); su_{min} = \min(0, X);$

    $lint_1^n = 0; rint_1^n = 1; nint^n = 1; pos = 1;$

    for $M \leftarrow 1$ **to** m **do**

      $lint^c = lint; rint^c = rint; nint^c = nint; nint = 0; sl = bl_{max};$

      for $j \leftarrow 1$ **to** $nint^c$ **do**

        $pos = pos + 2;$

        $i_l = lint_j^c;\ i_m = pos - 1;\ i_r = rint_j^c;\ t_{i_m} = \frac{t_{i_r} + t_{i_l}}{2};\ \delta = \frac{t_{i_r} - t_{i_l}}{2v};$

        $Y_p = Beta^{-1}(sobol_s(k, pos-1), \delta, \delta);$

        $\gamma_{i_m}^+ = \gamma_{i_l}^+ + Y_p(\gamma_{i_r}^+ - \gamma_{i_l}^+);$

        $Y_n = Beta^{-1}(sobol_s(k, pos), \delta, \delta);$

        $\gamma_{i_m}^- = \gamma_{i_l}^- + Y_n(\gamma_{i_r}^- - \gamma_{i_l}^-);$

        $su_{min} = min(su_{min}, \gamma_{i_m}^+ - \gamma_{i_m}^- + t_{i_m}\mu);$

        $i_n = [i_l, i_m]; i_p = [i_m, i_r];$

        for $w \leftarrow 1$ **to** 2 **do**

          $lb = \gamma_{i_n(w)}^+ - \gamma_{i_p(w)}^- + t_{i_n(w)}\mu + (t_{i_p(w)} - t_{i_n(w)})\mu^-;$

          $ub = \gamma_{i_p(w)}^+ - \gamma_{i_n(w)}^- + t_{i_n(w)}\mu + (t_{i_p(w)} - t_{i_n(w)})\mu^+;$

          if $lb \leq su_{min}$ **then**

            $nint^n = nint + 1; lint_{nint^n}^n = i_n(w); rint_{nint^n}^n = i_p(w);$

            $sl = \min(sl, lb);$

      if $e^{-rT} S_0(e^{su_{min}} - e^{sl})) \leq 2\epsilon$ **then**

        break

    $\text{Payoff}_N(k) = e^{-rT} S_0(e^X - \frac{1}{2}(e^{su_{min}} + e^{sl}));$

  $Call_n(p) = sum(\text{Payoff}_N)/N_{rQMC}$

$C = \frac{1}{n} sum(Call_n); se = \sqrt{var(Call_n)/n};$

The pseudocode for applying the rQMC to Algorithm 2 for pricing the lookback option under the VG model using Adaptive DGBS is shown above as Algorithm 4 for reference. According to Algorithm 4, lines 5 to 7 are to generate the shifted Sobol point set by Cranley-Patterson Rotation with size $(2 \times (2^m + 1))$ by N. In this case, we use $2 \times (2^m + 1)$ coordinates for each point u_i because we need at most 2^m randomized quasirandom numbers to generate each sequence of beta random variate by inversion in lines 18 and 20. The depth m is chosen to be sufficiently large as an upper bound to obtained the desired efficiency when using the Adaptive DGBS framework. The two extra shifted Sobol points generate the random gamma variates using inversion in line 9 for each stage M, $M \in \{1, \ldots, m\}$. As for the choice of $n \in \mathbb{N}$, i.e. the numbers of copies of Sobol point set, we take n equals 25 to obtain a reasonable estimate of the variance when evaluating the fair price for the floating strike call option price under the VG model. The other parts of Algorithm 4. are similar to Algorithm 2. except for some slight modifications on indexes in lines 15 and 16.

When calculating the option's fair price, the approach in the rQMC case is quite different from using the Monte-Carlo simulation, where in the Monte-Carlo case, we take the average of the payoff for the option at each simulation and obtain the fair price of the option. However, for the rQMC implementation, we calculate the payoff of the options using Adaptive DGBS at each simulation k in line 31 and take the average of the corresponding payoff to obtain the call price for the p^{th} copy of the Sobol sequence in line 32. As a result, we can obtain the estimated fair price of the floating strike lookback call option under the VG model in the Adaptive DGBS framework as

$$\begin{aligned} \hat{C}_{rQMC,\,F} &= \frac{1}{n}\sum_{p=1}^{n}\left(\frac{1}{N_{rQMC}}\sum_{k=1}^{N_{rQMC}} e^{-rT}S_0[e^{x_{k,p}} - \frac{1}{2}(e^{su_{min_{k,p}}} + e^{sl_{k,p}})]\right) \\ &= \frac{1}{n}\sum_{p=1}^{n}\mathrm{Call}_n(p) \end{aligned} \tag{5.3}$$

where $\mathrm{Call}_n(p)$ refers to the estimated price of the floating strike lookback call option at each p^{th} randomization (shift). The corresponding standard error for $\hat{C}_{rQMC,\,F}$ is

$$se_{rQMC,\,F} = \sqrt{\frac{\sum_{p=1}^{n}(\mathrm{Call}_n(p) - \hat{C}_{rQMC,\,F})^2}{n(n-1)}} \tag{5.4}$$

A similar argument can be applied to the estimation of the Barrier option in either a knock-in or a knock-out condition. In the next chapter, we will show how our Adaptive DGBS implementation using the rQMC method for the valuations of the lookback option and barrier option outperforms the Monte-Carlo simulation in terms of higher accuracy.

Chapter 6

Numerical Results

In this chapter, we will examine the performance of the Adaptive DGBS algorithm (Algorithm 1) and its applications on both the lookback and barrier options. Furthermore, we will compare the performance of the Monte-Carlo method with the rQMC method for pricing these options under the VG model with Adaptive DGBS framework. According to Avramidis and L'Ecuyer (2006)[2], they have fitted the VG model over the real-world market data and estimated the variance gamma model and options parameters. Specifically, the estimated parameters are summarized in Table 6.1 from below.

Table 6.1: Parameters for Numerical Results from Section 6.1 to 6.3

σ	0.1927	r	0.0548
v	0.2505	q	0
θ	-0.2859	S_0	100
T	0.40504	μ	0.31356

All algorithms were implemented in MATLAB (The Mathwork, Inc. (2019)[15]), using the built-in random number utility functions and quasirandom sequence package (*sobolset()* and *net()*). All computational results are running on Quad-Core of an Intel Core i5 2.4 GHz computer.

6.1 Simulating Final and Extremal values

In our first numerical experiments, we examine the efficient gain by the adaptive DGBS method for simulating triplet of final and extremal values for the VG process (Algorithm 1) to full-dimensional DGBS path sampling (Appendix A.2.1). The efficiency gain is measured by the ratios of the average number of simulated points in DGBS per path over the average number of simulated points in Adaptive DGBS per path. The average simulated points per path for the Adaptive DGBS is given by $\frac{1}{N}\sum_{k=1}^{N} 2^{d_k^{(i)}}$, where $d_k^{(i)}$ is the depth of the dyadic partition of the Adaptive DGBS stops at k^{th} simulations at a given tolerance level $\epsilon = 10^{-i}$ for the stopping criteria stated in Proposition 3.1.1, where we set $i \in \{2, 6, 10, 14\}$ and $N = 10^6$ triplet simulations. The numerical results for simulating the VG processes' final and extremal values using both Adaptive DGBS and DGBS are summarized in Table 6.2.

Table 6.2: Comparison of the Average Numbers of Simulated Points per Path for Adaptive DGBS Versus Full-Dimensional DGBS Sampling with Depth $m = 10$

Tolerance level ϵ	10^{-2}	10^{-6}	10^{10}	10^{-14}
Adaptive DGBS	94.77	159.31	179.58	207.99
DGBS	2^{10}	2^{10}	2^{10}	2^{10}
Efficiency gained	10.81	6.43	5.70	4.92

According to Table 2, there is large efficiency gained for using the Adpative DGBS algorithm in simulating the final and extremal values of the VG processes. It is also worthy to note that for a reasonable error bound $\epsilon = 10^{-6}$ of the approximation error, the Adaptive DGBS algorithm generated 10,000 triplets of the final and extremal values per second through the Monte-Carlo simulations, which is considerably faster than when using DGBS.

6.2 Valuation of Lookback Option

In our second numerical analysis, we apply the Adaptive DGBS for evaluating the fair price of a floating strike lookback option under the VG model by setting a stopping condition, according to Section 4.1, with prior error bounds $\epsilon = 10^i$, for $i \in \{0, -2, -4, -6, -8, -10\}$. In particular, for each choice of maximum depth, we choose a sufficiently large value of

m, i.e. $m = 8$, to achieve the desired efficiency of using the Adaptive DGBS algorithm. Moreover, other than only using the classical Monte-Carlo method, we also applied the rQMC method explained in Chapter 5 by using $n = 25$ copies of low discrepancy Sobol sequence with a Cranley-Patterson Rotation shift to reduce the estimation variance. To accommodate the usage of rQMC over a dyadic partition time grid in quickly producing the Sobol set, we typically use the power of 2 inner simulation times, i.e. for MC simulation, we use $N = 25 \times 2^{16}$, while for rQMC simulation, we use $N_{rQMC} = 2^{16}$. Consequently, the number of total simulations $N = 25 \times 2^{16} = N_{rQMC} \times n$ is consistent for both simulation methods. The other parameters are specified in Table 6.1. The simulation results for pricing the floating strike lookback option under the VG model using Adaptive DGBS in both MC simulation and the rQMC simulation are summarized in Table 6.3 below.

Table 6.3: Comparison of MC Versus rQMC on Pricing the Floating Strike Lookback Call Option Under VG Model with the Adaptive DGBS framework

Tolerance level ϵ	10^{0}	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}
MC						
Estimated Price	9.5559	9.3417	9.3513	9.3442	9.3409	9.3406
Standard Error	0.00565	0.00560	0.00561	0.00560	0.00561	0.00561
CPU time(sec)	49.1	62.0	62.5	61.4	61.6	61.3
rQMC						
Estimated Price	9.5722	9.3970	9.3973	9.3982	9.3983	9.3978
Standard Error	$5.93{\cdot}10^{-4}$	$5.90{\cdot}10^{-4}$	$6.24{\cdot}10^{-4}$	$6.63{\cdot}10^{-4}$	$6.28{\cdot}10^{-4}$	$4.74{\cdot}10^{-4}$
CPU time(sec)	172.4	605.1	619.5	621.3	633.0	639.5

According to Table 6.3, for the same total numbers of simulations at each tolerance level ϵ, both simulation methods with adaptive DGBS algorithm provides accurate estimates of the fair price for the floating strike lookback call option under the VG model. In particular, the rQMC outperforms the MC simulation in terms of lower standard error in estimating the price of the floating strike lookback call option under the VG model within the Adaptive DGBS framework. The standard errors incurred by estimating the option with rQMC are about 10 times smaller than using MC simulation on average. Thus, in this case, the rQMC method works well when applied to Algorithm 2 and it gives a significant reduction of estimation error when pricing the floating strike lookback call option under the VG processes within the Adaptive DGBS framework. However, due to the slow speed of generating random numbers by inversion for both random gamma and beta variates and

generating large dimensional Sobol sets, the computational efforts for using the rQMC are 10 times higher than the one using MC simulation on average. According to Avramidis and L'Ecuyer (2006)[2], the fair price of the floating strike lookback call option under the VG model is 9.39805±0.00015 with 95% confidence. Thus, with the same total numbers of simulations and tolerance level ϵ, the rQMC gives more accurate results of the option's fair price than using MC. For both MC and rQMC methods, in general, the lower the tolerance level ϵ, the more accurate the option price will be, and the more computational times both simulation needs in estimating the fair price of the lookback option at each tolerance level. Generally, getting small standard errors for estimating the price of an option is not the only goal, Avramidis and L'Ecuyer (2006)[2] have also included the minimization of the mean square error (MSE) for the option price estimator with further optimization of the pricing problem. However, this is not our research's goal. Instead, one can read the paper [2] for further details of the approach.

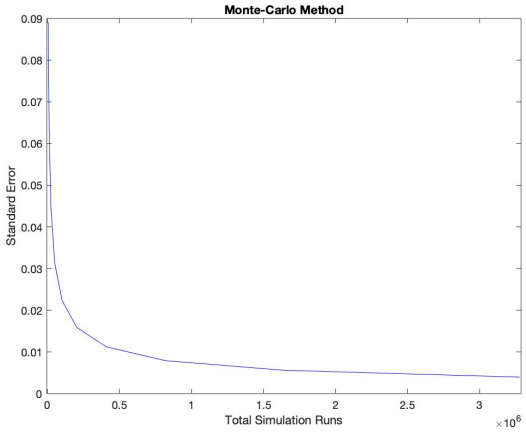
6.2.1 Sensitivity Analysis

In this section, we will perform the sensitivity analysis to see how the variables of our focus are affected based on changes in other input variables. Specifically, we want to investigate how changes among the input variables: simulation runs N for MC, simulation depth m for the VG processes, and the inner simulation runs N_{rQMC} for rQMC will affect the standard error of the estimated floating strike lookback call option's price and the estimated option price itself.

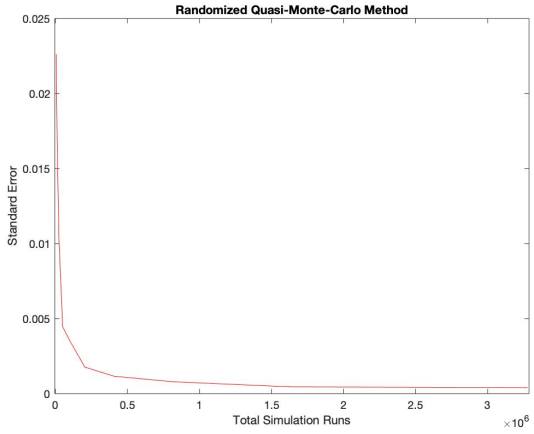
In our first analysis, we want to see how the number of simulations would affect the estimated standard error of the floating strike lookback call option estimator under the VG model within the Adaptive framework in both MC and rQMC cases. Specifically, we fixed the simulation depth m for the VG processes and increased both the simulation N runs for MC and simulation runs N_{rQMC} for rQMC. In particular, we fixed the simulation depth $m = 8$, the tolerance level $\epsilon = 10^{-6}$, and the number of copies of Sobol sequence $n = 25$. We further let $N \in \{25 \cdot 2^8, 25 \cdot 2^9, \ldots, 25 \cdot 2^{16}, 25 \cdot 2^{17}\}$ and $N_{rQMC} \in \{2^8, 2^9, \ldots, 2^{16}, 2^{17}\}$. The remaining parameters are the same as in Table 6.1.

According to Figure 6.1, the standard errors of the Adaptive DGBS estimator for floating strike lookback call option under the VG model using both MC and rQMC method within the Adaptive DGBS framework show convergence to zeros. This finding implies that both the MC and the rQMC estimators converge to the actual value of the floating strike lookback call option under the VG model as the number of simulations goes up. In this case, we give the linear fits of the lines for implementing the rQMC and MC in Figure

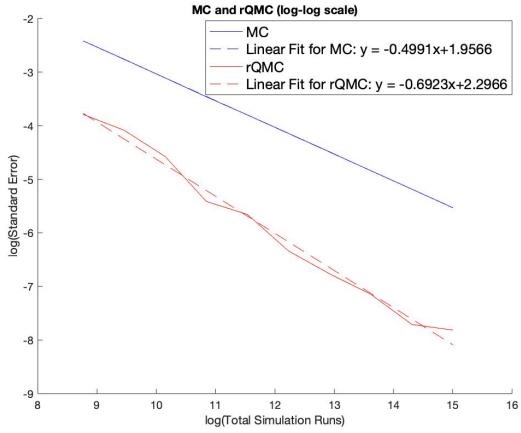
6.1 (c) by using the polynomials of degree 1 that provide the best fit for the standard error with package *polyfit()* in MATLAB. In particular, according to Figure 6.1(c): the log-log plot for both MC and rQMC method, the convergence rate of using the rQMC method is significantly faster than using the MC with a steeper slope, i.e. the slope of rQMC $\approx -0.6923 < -0.4991 \approx$ the slope of MC. More importantly, the standard error of using MC has constant convergence rates over the total simulations, whereas the standard error of using the rQMC has non-constant convergence rates, as indicated in Figure 6.1 (c).



(a) Monte-Carlo Method



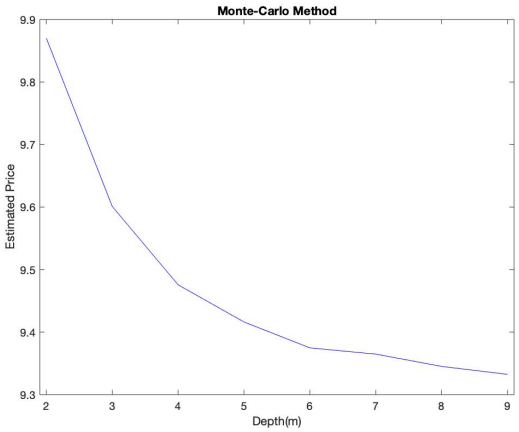
(b) Randomized Quasi-Monte-Carlo Method



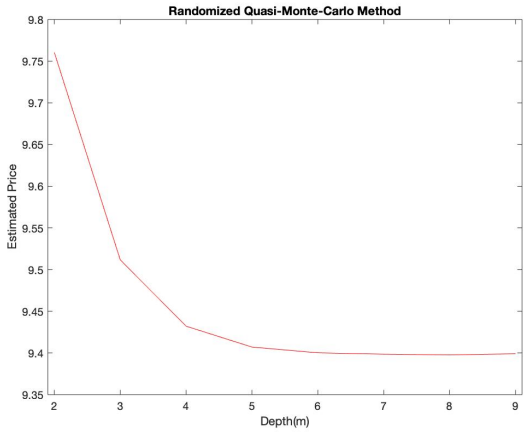
(c) MC and rQMC in Log Scale

Figure 6.1: Sensitivity Analysis on the Estimated Standard Error of Lockback Call Estimator under VG Model with Adaptive DGBS for Various $N\&N_{rQMC}$

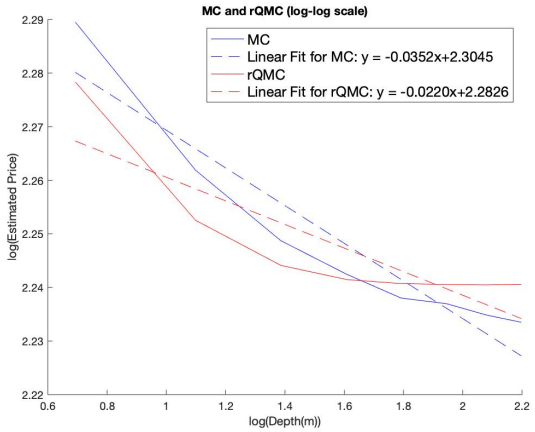
Moreover, as previously mentioned, for the same total numbers of simulations at each tolerance level ϵ, the rQMC simulations give more accurate estimated price of the option than using MC simulations in terms of significant smaller standard errors as Figure 6.1 indicates.



(a) Monte-Carlo Method



(b) Randomized Quasi-Monte-Carlo Method



(c) MC and RQMC in Log Scale

Figure 6.2: Sensitivity Analysis on the Estimated Price of Lockback Call Option under VG Model with Adaptive DGBS for Various Depth m

In our second analysis, we want to know how the number of depth will affect the estimated price of the floating strike lookback call option under the VG model within the

Adaptive DGBS framework in both the MC and rQMC cases. In this case, we fixed the number of simulation runs $N = 25 \cdot 2^{16}$ for MC simulation, the number of inner simulation runs $N_{rQMC} = 2^{16}$ for rQMC simulation, the tolerance level to $\epsilon = 10^{-6}$, and initialized the number of copies of Sobol sequence with a rotation shift to $n = 25$. We then consider an increasing set of depth $m \in \{2, 3, \ldots, 8, 9\}$ and the remaining parameters are the same as in Table 6.1.

The numerical results are shown in Figure 6.2. The x coordinate refers to the number of depth m, and the y coordinate refers to the estimated price of the lookback call option under the VG model within the Adaptive DGBS framework. According to Figure 6.2, as the number of depth m goes up, both the MC and the rQMC estimators for floating strike lookback call option within the Adaptive DGBS framework show convergence to the actual value of the floating strike lookback call option.

The same as our first analysis, we approximated the lines of using both MC and rQMC in Figure 6.2 (c) by the polynomial of degree 1 that best fit the estimated prices of the lookback call option under the VG model within the Adaptive DGBS framework. In particular, based on Figure 6.2 (c): the log-log plot of MC and rQMC, the convergence rate of the estimator using the rQMC method to the actual value of the option is slower than using the MC with a smoother slope, i.e. the slope of the rQMC $\approx -0.0220 > -0.0352 \approx$ the slope of MC. Furthermore, the estimator using the rQMC method converges to the true value after depth $m = 6$ and keep the values the same as the true value when increasing the depth m. (Figure 6.2 (b)) In contrast, the estimator using the MC method converges to the true value after depth $m = 5$, but with values continue to drop as the depth m increases (Figure 6.2 (a)). In this case, both estimated prices of the lookback call option under the VG model within the Adaptive DGBS using the MC and rQMC have non-constant convergence rates over the depth m, as indicated in Figure 6.2 (c)

6.3 Valuation of Barrier Option

In the last numerical analysis, the valuation of the barrier call option under the VG model within the Adaptive DGBS framework is investigated using both Monte-Carlo simulation and randomized Quasi-Monte-Carlo simulation. (see Algorithm 3 and Algorithm 4) Different than estimating the fair price of the floating strike lookback option under the VG model, if we set the maximum depth to be sufficiently large, i.e. let $m = 8$ in this case, the payoff's bounds of the option will eventually converge to a single quantity. Thus, the breaking condition for a given tolerance level ϵ is not needed for pricing the barrier call option under the VG model within the Adaptive DGBS framework. Again, according to

Avramidis and L'Ecuyer (2006) [2], they have fitted the model into real market data and the corresponding estimated parameters of the variance gamma process and options are given in the following Table 6.4.

Table 6.4: Parameters for Numerical Results in Section 6.4

σ	0.19071	r	0.0549
v	0.49083	q	0.011
θ	-0.28113	S_0	100
T	0.46575	μ	0.31356
m	8	K	100

In this case, we use $N = 25 \cdot 2^{16}$ simulations for the MC method, $N_{rQMC} = 2^{16}$ for the inner simulation of rQMC method, and $n = 25$ copies of the Sobol sequence with Cranley-Patterson rotation shift to ensure consistency for the total number of simulation between the MC and the rQMC method. Upon various barrier options, we are typically interested in pricing the up-and-in call and down-and-out call option. Thus, we set the lower barrier $B_l \in \{80, 85, 90\}$ for down-and-out call option and the upper barrier $B_u \in \{105, 110, 120\}$ for up-and-in call option. Table 6.5 shows results in estimating the price of various single barrier options under the VG model within the Adaptive DGBS framework using both MC and rQMC simulation methods for different barriers.

Table 6.5: Comparison of Results and Speed of Barrier Option Estimation with Adaptive DGBS (Various Single Barrier Call Options) Using MC Versus Using RQMC

	Up&In B_u**=105**	**Up&In** B_u**=110**	**Up&In** B_u**=120**	**Down&Out** B_l**=80**	**Down&Out** B_l**=85**	**Down&Out** B_l**=90**
MC						
Estimated Price	7.2180	6.1509	1.9763	7.4553	7.2707	6.5329
Standard Error	0.0065	0.0068	0.0055	0.0064	0.0064	0.0065
CPU Time(sec)	14.45	14.79	12.77	10.57	10.60	10.40
rQMC						
Estimated Price	7.3921	6.5678	2.1625	7.4547	7.2839	6.5288
Standard Error	$1.91 \cdot 10^{-4}$	$5.66 \cdot 10^{-4}$	$7.51 \cdot 10^{-4}$	$1.70 \cdot 10^{-4}$	$1.70 \cdot 10^{-4}$	$1.78 \cdot 10^{-4}$
CPU Time(sec)	68.91	76.18	49.41	48.18	45.45	44.83

According to Table 6.5, both MC and rQMC methods yield accurate estimations for the barrier option's fair prices under the VG processes within the Adaptive DGBS framework with small standard errors for the estimators. In particular, under the same total

simulation runs and depth, the rQMC method gives a smaller standard error in estimating the barrier option's fair price than using the MC simulation. However, as compensation for this accuracy, the random number generations using the inverse function with numerical treatments that potentially involve the expansion of polynomials will slow down the rQMC simulation. In particular, the rQMC method generally takes 4-5 times longer computational time on average in pricing the barrier option under the VG model with the Adaptive DGBS than the MC method, according to Table 6.5.

Notice that as the barrier B_l increases to near 100 for the down-and-out call option with strike price $K = 100$ and underlying spot price $S_0 = 100$, the option has a lesser value. These observations are in accordance with the definition of a down-and-out call option. As the underlying price reaches the lower barrier, the option becomes worthless. Similar observations can be made on the estimated price of the up-and-in call option under the VG model within the Adaptive DGBS in both the MC and the rQMC cases. In particular, as B_u increases its departure from the underlying spot price $S_0 = 100$, the up-and-in call option with strike $K = 100$ will have a smaller value due to the structure of the up-and-in call option. Moreover, according to Table 6.5, for both MC and rQMC simulations methods, the closer the lower barrier B_l is to the spot price of the underlying asset $S_0 = 100$, the faster the simulations will take to estimate the fair price of the down-and-out option under the VG model in general.

On the other hand, the observations from the running times of the up-and-in call option case in Table 6.5 are also reasonable, where for both MC and rQMC simulations, as B_u departs from the underlying spot price $S_0 = 100$, $CPU_{B_u=120} < CPU_{B_u=105} < CPU_{B_u=110}$. In interpreting these observations, the lower the B_u, the faster will be the simulation due to the ease of early crossing of the simulating point to the barrier B_u. On the other hand, for higher B_u, since the upper bound for the simulated underlying price under the VG model will always stay far below the upper barrier B_u and many of the paths might not pass line 9 condition in Algorithm 3. As a result, the payoff of the option can be determined in lesser iterations. Thus, this fact leads to a faster simulation for the higher barrier B_u than the smaller barrier B_u. Eventually, for the mid-barrier B_u, it is essentially the combination of the higher barrier B_u and the smaller barrier B_u. After a few steps of Adaptive DGBS, some simulated paths might not pass the line 9 condition, and some might pass above the barrier. Thus, according to line 28 in Algorithm 3, those who pass the barrier might still be indefinite for categorizing as the knock-in case. Thereby more simulations are needed. Therefore, in general, the mid-barrier B_u will certainly need a longer time to determine the payoff than the higher barrier B_u and the smaller barrier B_u. Through above arguments, the inequality $CPU_{B_u=120} < CPU_{B_u=105} < CPU_{B_u=110}$ is correct.

Chapter 7

Conclusion

In this paper, we reviewed the general concept of the variance gamma (VG) process, a pure jump Lévy process that has independent and stationary paths. In sampling the VG process, the difference-of-gamma sampling (DGBS) is introduced using the difference-of-gamma representation of the VG process with two independent gamma processes. For risk-neutral pricing purposes, we add a drift term into the original VG process. Later, to efficiently price the options under the VG model, we introduced and developed a fast and unbiased approach for Monte-Carlo valuation of the lookback and barrier options using the Adaptive difference-of-gamma sampling method over a dyadic partition time grid, which was proposed by Becker (2010)[4]. This Adaptive difference-of-gamma sampling method dynamically rules out the dispensable paths that neither contributes to the minimal nor the maximal values of the process when sampling the underlying asset prices under the VG process. The typical features of the Adaptive DGBS provided advantages in pricing the lookback and barrier option that relies on the extremal values of the underlying prices under the VG model. Besides, we have implemented the randomized Quasi-Monte Carlo Method to evaluate these options under the VG model within the Adaptive DGBS framework by a low discrepancy Sobol set in base 2 with a Cranley-Patterson Rotation shift.

The numerical experiments show that the Adaptive DGBS algorithm gives significant computational saving to generate the VG processes' final and extremal values compared to the original DGBS method. For valuation of the path-dependent options, the Adaptive DGBS method using either Monte-Carlo or randomized Quasi-Monte-Carlo simulation gives a fast and accurate estimation of the fair price for the lookback and barrier options under the VG model. In particular, rQMC is more accurate and computationally costly than the MC in pricing those options. Other variance reduction techniques like importance sampling can also be applied under this framework as a potential improvement.

References

[1] Dilip B. Madan Ali Hirsa. Pricing American options under variance gamma. *Journal of Computational Finance*, (7):63–80, 2004.

[2] Athanassios Avramidis and Pierre L'Ecuyer. Efficient Monte Carlo and Quasi-Monte Carlo option pricing under the variance gamma model. *Management Science*, 52:1930–1944, 12 2006.

[3] Athanassios Avramidis, Pierre L'Ecuyer, and Pierre-Alexandre Tremblay. Efficient simulation of gamma and variance-gamma processes. *Winter Simulation Conference Proceedings*, 1:319–326 Vol.1, 01 2004.

[4] Martin Becker. Unbiased Monte Carlo valuation of lookback, swing and barrier options with continuous monitoring under variance gamma models. *The Journal of Computational Finance*, 13:35–61, 06 2010.

[5] Jean Bertoin. *Lévy Processes*. Cambridge University Press, Cambridge, UK, 1998.

[6] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.

[7] Rama Cont and Peter Tankov. *Financial Modelling With Jump Processes*. Financial Mathematics Series Chapman and Hall/CRC, Boca Raton FL, London, UK, 2004.

[8] Rama Cont and Ekaterina Voltchkova. A finite difference scheme for option pricing in jump diffusion and exponential Lévy models. *SIAM Journal on Numerical Analysis*, 43(4):1596–1626, 2006.

[9] Luc Devroye. *Non-Uniform Random Variate Generation*. Springer, Montreal, Canada H3A 2K6, first edition, 1986.

[10] Luc Devroye. Random variate generation in one line of code. In *Proceedings of the 28th Conference on Winter Simulation*, WSC '96, pages 265–272, USA, 1996. IEEE Computer Society.

[11] François Dufresne, Hans U. Gerber, and Elias S. W. Shiu. Risk theory with the gamma process. *ASTIN Bulletin*, 21(2):177–192, 1991.

[12] Christiane Lemieux. *Monte Carlo and Quasi-Monte Carlo sampling*. Springer-Verlag, New York, US, 2009.

[13] Dilip B. Madan, Peter P. Carr, and Eric C. Chang. The variance gamma process and option pricing. *Review of Finance*, 2(1):79–105, 04 1998.

[14] Dilip B. Madan and Eugene Seneta. The variance gamma (v.g.) model for share market returns. *The Journal of Business*, 63(4):511–524, 1990.

[15] The Mathworks, Inc., Natick, Massachusetts. *MATLAB version 9.7.0.1319299 (R2019b) Update 5*, 2019.

[16] Ken-iti Sato. *Lévy Processes and Infinitely Divisible Distributions*. Cambridge University Press, Cambridge, UK, 1999.

APPENDICES

Appendix A

Pseudocode & Figure

A.1 Pseudocode for Sampling Gamma Process

A.1.1 Gamma Bridge Sampling (GSS) of Gamma Process

Algorithm 5: GSS of a Gamma Process with Parameter (μ,v) for a 2^{-m}-point Equal Length Partition of [0,T] by Avarmidis (2004)[3]

input : $\mu = 1$, $v > 0$, Final time $T > 0$, depth of the partition $m \in \mathbb{N}$, $t_0 = 0$

output: A realization of the gamma process $\gamma_t^{\mu=1,v}$ on $t \in [0, T]$

$\gamma_{t_0}^{\mu=1,v} = 0$;
$h = 2^{-m}T$;
for $i \leftarrow 1$ **to** 2^m **do**
    $Increment =$ random $\Gamma(\mu^2 h/v, v/\mu)$;
    $\gamma_{ih} = \gamma_{(i-1)h} + Increment$;

A.1.2 Gamma Bridge Sampling (GBS) of Gamma Process

Algorithm 6: GBS of a Gamma Process with Parameter (μ,v) for a 2^{-m}-point Equal Length Partition of [0,T] by Avarmidis (2004)[3]

input : $\mu = 1$, $v > 0$, Final time $T > 0$, depth of the partition $m \in \mathbb{N}$, $t_0 = 0$

output: A realization of the gamma process $\gamma_t^{\mu=1,v}$ on $t \in [0, T]$

$\gamma_{t_0=0}^{\mu=1,v} = 0$;

$\gamma_{t_{2^m}=T}^{\mu=1,v} = random\ \Gamma(\mu^2 T/v, v/\mu)$;

for $l \leftarrow 1$ **to** m **do**

  for $m \leftarrow 1$ **to** 2^{l-1} **do**

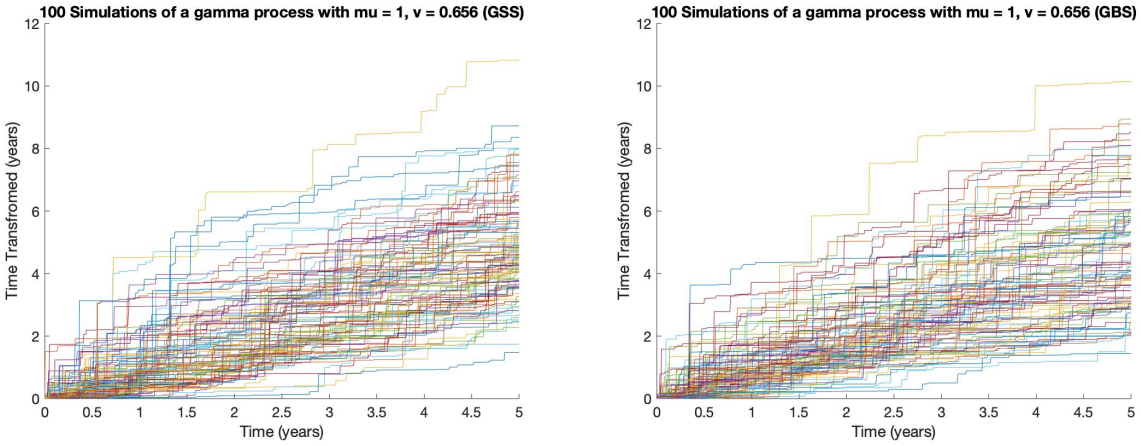
    $i = 2m - 1$;

    $\delta = \mu^2 T/(v2^l)$;

    Y = random $Beta(\delta, \delta)$;

    $\gamma_{i2^m/2^l} = \gamma_{(i-1)2^m/2^l} + \mathrm{Y}(\gamma_{(i+1)2^m/2^l} - \gamma_{(i-1)2^m/2^l})$;

A.1.3 Figures of Sampling Gamma Process



(a) Simulations of a gamma process using GSS (b) Simulations of a gamma process using GBS

Figure A.1: Sampling Gamma Process on a 2^{-m} Points Equal-Length Partition on [0,T]

A.2 Pseudocode for Sampling Variance Gamma Process

A.2.1 Difference of Gamma Bridge Sampling (DGBS) of VG

Algorithm 7: DGBS of a Variance Gamma Process with Parameter (μ,θ,σ,v) for a 2^{-m}-point Equal Length Partition of [0,T] by Avarmidis (2004)[3]

input : Variance Gamma process parameters μ, $\theta \in \mathbb{R}$, σ, $v > 0$, terminal time T, depth of the dyadic partition $m \in \mathbb{N}$, $t_0 = 0$

output: A realization of the Variance Gamma process $X_t^{(\mu,\theta,\sigma,v)}$ on $t \in [0, T]$

$\mu_p = (\sqrt{\theta^2 + 2\sigma^2/v} + \theta)/2;\ \mu_n = \mu_p - \theta;$
$v_p = \mu_p^2 v,\ v_n = \mu_n^2 v;$
$\gamma^+_{t_0=0} = 0;\ \gamma^-_{t_0=0} = 0;$
$\gamma^+_{t_{2^m}=T} = random\ \Gamma(\mu_p^2 T/v_p, v_p/\mu_p);$
$\gamma^-_{t_{2^m}=T} = random\ \Gamma(\mu_n^2 T/v_n, v_n/\mu_n);$
$t = linspace(0, T, 2^m);$
for $l \leftarrow 1$ **to** m **do**
  for $m \leftarrow 1$ **to** 2^{l-1} **do**
    $i = 2m - 1;$
    $\delta_p = \mu_p^2 T/(v_p 2^l);$
    $Y_p = \text{random } Beta(\delta_p, \delta_p);$
    $\gamma^+_{i2^m/2^l} = \gamma^+_{(i-1)2^m/2^l} + Y_p(\gamma^+_{(i+1)2^m/2^l} - \gamma^+_{(i-1)2^m/2^l});$
    $\delta_n = \mu_n^2 T/(v_n 2^l);$
    $Y_n = \text{random } Beta(\delta_n, \delta_n);$
    $\gamma^-_{i2^m/2^l} = \gamma^-_{(i-1)2^m/2^l} + Y_n(\gamma^-_{(i+1)2^m/2^l} - \gamma^-_{(i-1)2^m/2^l});$
    $X_{iT/2^l} = \gamma^+_{i2^m/2^l} - \gamma^-_{i2^m/2^l};$
$X = X + \mu t;$