Pricing Bermudan Swaptions on the LIBOR Market Model using the Stochastic Grid Bundling Method

Stef Maree[*],
Jacques du Toit[†]

Abstract

We examine using the Stochastic Grid Bundling Method (SGBM) to price a Bermudan swaption driven by a one-factor LIBOR Market Model (LMM). Using a well-known approximation formula from the finance literature, we implement SGBM with one basis function and show that it is around six times faster than the equivalent Longstaff–Schwartz method. The two methods agree in price to one basis point, and the SGBM path estimator gives better (higher) prices than the Longstaff–Schwartz prices. A closer examination shows that inaccuracies in the approximation formula introduce a small bias into the SGBM direct estimator.

1 Introduction

The LIBOR Market Model (LMM) is an interest rate market model. It owes much of its popularity to the fact that it is consistent with Black's pricing formulas for simple interest rate derivatives such as caps and swaptions, see, for example, [3].

Under LMM all forward rates are modelled as individual processes, which typically results in a high dimensional model. Consequently when pricing more complex instruments under LMM, Monte Carlo simulation is usually used.

When pricing products with early-exercise features, such as Bermudan swaptions, a method is required to determine the early-exercise policy. The most popular approach to this is the Longstaff–Schwartz Method (LSM) [8]. When time is discrete (as is usually the case in numerical schemes) Bellman's backward induction principle says that the price of the option at any exercise time is the maximum of the spot payoff and the so-called continuation value. In LSM this continuation value can be approximated through regression: a set of basis functions is chosen which are closely correlated with the option value, and these are then regressed on the option values at the next time step to get a simple functional form for the continuation value. From the continuation value it is easy to determine the optimal exercise policy.

LSM is an intuitive method which is easy to implement. Due to its generality, it is also widely applicable. However, many sample paths are often required to generate a suitably small confidence interval for the option value, which means that accurate pricing with LSM can be slow. The Stochastic Grid Bundling Method (SGBM) introduced in [5] is a competitor to LSM. While SGBM is also based on regression, it makes different approximations and requires more analytic inputs. The result is a method which has much lower variance, but is often difficult to implement for more complex underlyings since the necessary analytic inputs are frequently not available.

To date SGBM has been applied to geometric Brownian motion (see [6]), to the Heston model, and to a Heston–Hull–White process (see [1]). In all these cases it compared

[*]Delft University of Technology, email: s.c.maree@student.tudelft.nl
[†]Numerical Algorithms Group, email: jacques@nag.co.uk

favourably to LSM. The aim in this paper is to see whether SGBM can be applied to the LIBOR Market Model and how the results compare to LSM. We chose to price a Bermudan swaption since this is a relatively simple path dependent product which is still quite widely traded.

2 Bermudan Pricing Problem

Consider a finite time horizon $[0, T]$ and let $\mathbf{W} = (\mathbf{W}(t))_{t\geq 0}$ be a d-dimensional Brownian motion under the risk-neutral measure $\mathbb{P}$. Let $\big(B(t)\big)_{t\geq 0}$ denote a risk-less bank account and $\big(S(t)\big)_{t\geq 0}$ a risky asset, and define a fixed set of times $\mathcal{T} := \{T_m\}_{m=0}^{M}$, where $0 = T_0 < T_1 < \cdots < T_M = T$. Let $h(S(T_m))$ be the intrinsic value of a Bermudan option, i.e. the holder of the option receives $\max\big(h(T_m), 0\big)$ if the option is exercised at time T_m.

The value $V(T_0, S(T_0))$ of the Bermudan option at time T_0 is then given by

$$V(T_0, S(T_0)) = \max_{\tau\in\mathcal{T}} \mathbb{E}\left[\frac{h(S(\tau))}{B(\tau)}\right], \tag{1}$$

where τ is a stopping time taking values in $\mathcal{T}$. To find the option price at time T_0 we apply backward induction. Starting from the option value at maturity T_M, we know the option value is equal to the payoff

$$V\big(T_M, S(T_M)\big) = \max\big(h(S(T_M)), 0\big). \tag{2}$$

For any other time T_m with $0 \leq m < M$ we assume by induction that $V(T_{m+1}, S(T_{m+1}))$ is known. If we define the *continuation value* at time T_m by

$$Q(T_m, S(T_m)) := \mathbb{E}\left[\frac{B(T_m)}{B(T_{m+1})} V(T_{m+1}, S(T_{m+1})) \;\middle|\; S(T_m)\right], \tag{3}$$

then the value of the Bermudan option at time T_m is given by

$$V(T_m, S(T_m)) = \max\big(h(S(T_m)), Q(T_m, S(T_m))\big), \tag{4}$$

if the option is still alive at time T_m. The main challenge is the computation of the continuation value in (3), which is typically accomplished by the LSM and is based on regression.

2.1 Bermudan Option Pricing with Longstaff–Schwartz (LSM)

The LSM [8] is a Monte Carlo based method to price Bermudan style options. As in any Monte Carlo method, a set of N paths is generated, $\{S^p(T_m) : p = 1, \ldots, N; m = 0, \ldots, M\}$, all starting from the same value $S(0) = S_0$. For each sample path p the option value at maturity $V(T_M, S^p(T_M))$ can be computed by (2).

Assume now that the set of option values $V(T_{m+1}, S^p(T_{m+1}))$, for $p = 1, \ldots, N$, is known. To compute the continuation value at time T_m, we regress the discounted option values at time T_{m+1} on a set of basis functions $\phi_k(T_m, S(T_m))$, for $k = 1, \ldots, K$, where K is a fixed (preferably small) number. In the language of simple linear regression, at each time step T_m the *dependent* variables are given by

$$y_p = \frac{B(T_m)}{B(T_{m+1})} V(T_{m+1}, S^p(T_{m+1})), \tag{5}$$

the *independent* variables are given by

$$x_{pk} = \phi_k(T_m, S^p(T_m)), \tag{6}$$

the regression parameters are β_k and the independent error is ε_p, resulting in the standard regression equation

$$y_p = \sum_{k=1}^{K} \beta_k x_{pk} + \varepsilon_p, \tag{7}$$

for $p = 1, \ldots, N$, which is readily solved.

One of the key assumptions in LSM is that at any time T_m we have

$$\frac{B(T_m)}{B(T_{m+1})} V(T_{m+1}, S(T_{m+1})) \approx \sum_{k=1}^{K} \beta_k \phi_k(T_m, S(T_m)), \tag{8}$$

where the β_ks were determined from (7). This functional form involves the stochastic process at time T_{m+1} and at time T_m. The SGBM method makes a slightly different assumption, as will be discussed later. Note that the parameters β_k are clearly dependent on time T_m.

Using (3) and (8) we can approximate the continuation value for each path by

$$\begin{aligned} Q(T_m, S^p(T_m)) &= \mathbb{E}\left[\frac{B(T_m)}{B(T_{m+1})} V(T_{m+1}, S(T_{m+1})) \;\middle|\; S^p(T_m) \right] \\ &\approx \mathbb{E}\left[\sum_{k=1}^{K} \beta_k(T_m) \phi_k(T_m, S(T_m)) \;\middle|\; S^p(T_m) \right] \\ &= \sum_{k=1}^{K} \beta_k(T_m) \phi_k(T_m, S^p(T_m)) \,. \end{aligned} \tag{9}$$

Once we have the continuation value at time T_m, we can compute the option value at T_m by (4) as

$$V(T_m, S^p(T_m)) = \max\left(h(S^p(T_m)),\ Q(T_m, S^p(T_m)) \right). \tag{10}$$

To be specific, not all sample paths $p = 1, \ldots, N$ are considered in (7) at each time step, only the paths that are in the money. If a sample path is out of the money, no choice has to be made, and the option value is just the discounted option value at the next time step [8].

While LSM is intuitive and easy to implement, there are some drawbacks. The main problem is the assumption (8) that LSM uses to bridge the time difference from T_{m+1} to T_m. It says that a function of the process at the next time step is a linear combination of basis functions of the process at the current time step. There are, therefore, two approximations here: the one saying that V can be approximated by a linear combination of K basis functions (this is the standard regression assumption) and the second saying that $S(T_{m+1})$ can be approximated by $S(T_m)$. It was shown in [4] that this time difference is a big source of randomness, and many sample paths are needed to reduce the variance suitably.

2.2 Bermudan Option Pricing with the Stochastic Grid Bundling Method (SGBM)

The SGBM differs from LSM in the way it handles the approximation (8). It clearly separates out the two assumptions that LSM combines.

We follow the construction in [1]. By backward induction we can assume that the option values $V(T_{m+1}, S^p(T_{m+1}))$ are known for all sample paths p. Suppose we are now at time T_m. SGBM begins by constructing a regression approximation to V at time T_{m+1}: it sets

$$y_p = V(T_{m+1}, S^p(T_{m+1})), \quad \text{and} \quad x_{pk} = \phi_k(T_{m+1}, S^p(T_{m+1})), \tag{11}$$

and solves the regression equation

$$y_p = \sum_{k=1}^{K} \beta_k x_{pk} + \varepsilon_p, \tag{12}$$

where the regression parameters β_k clearly depend on time T_{m+1}. This means that SGBM makes the approximation

$$V(T_{m+1}, S^p(T_{m+1})) \approx \sum_{k=1}^{K} \beta_k(T_{m+1})\phi(T_{m+1}, S^p(T_{m+1})), \tag{13}$$

for all sample paths p. Note that the parameters $\beta_k(T_{m+1})$ are determined during the iteration (backward induction) at time T_m, i.e. the $\beta_k(T_M)$'s are determined during the iteration at time T_{M-1} and so forth.

2.2.1 Projection

The biggest difference between SGBM and LSM is in the projection from time T_{m+1} to time T_m. Using (13) we compute the continuation value at time T_m as

$$\begin{aligned} Q(T_m, S^p(T_m)) &= \mathbb{E}\Big[\frac{B(T_m)}{B(T_{m+1})} V(T_{m+1}, S(T_{m+1}) \;\Big|\; S^p(T_m)\Big] \\ &\approx \mathbb{E}\Bigg[\frac{B(T_m)}{B(T_{m+1})} \sum_{k=1}^{K} \beta_k(T_{m+1})\phi_k\big(T_{m+1}, S(T_{m+1})\big) \;\Bigg|\; S^p(T_m)\Bigg] \\ &= \sum_{k=1}^{K} \beta_k(T_{m+1}) \mathbb{E}\Big[\frac{B(T_m)}{B(T_{m+1})} \phi_k\big(T_{m+1}, S(T_{m+1})\big) \;\Big|\; S^p(T_m)\Big] . \end{aligned} \tag{14}$$

Setting $\psi_k(T_m, S^p(T_m)) = \mathbb{E}\Big[\frac{B(T_m)}{B(T_{m+1})} \phi_k(T_{m+1}, S(T_{m+1})) \;\Big|\; S^p(T_m)\Big]$ we get

$$Q(T_m, S^p(T_m)) = \sum_{k=1}^{K} \beta_k(T_{m+1}) \psi_k\big(T_m, S^p(T_m)\big) . \tag{15}$$

The value of the option at time T_m is then given by (4) as before. Clearly the choice of basis functions for SGBM is more involved than for LSM, as they not only have to be representative for the option value, but we also require an expression for their discounted conditional expectation ψ_k.

2.2.2 Bundling

SGBM improves the regression approximation by grouping sample paths into bundles and regressing only within a bundle. The intuition here is simple: a Taylor series approximation of given order is most accurate close to the point where it was constructed. In a similar way, if we fix a (small) interval $\mathcal{B}$ and perform the regression (12) only for those $x_{pk} \in \mathcal{B}$, then we would expect a better fit to the data within $\mathcal{B}$ than if we had regressed across the whole space (see [1]). There is a caveat though, in that, there needs to be a sufficient number of sample paths in $\mathcal{B}$ for the regression to accurately capture the shape of the option value V. Note that the regression parameters β_k will now be dependent on $\mathcal{B}$: each bundle will have its own set of regression parameters so that we construct a set of local approximations.

At each time step T_m, SGBM constructs X_m bundles $\mathcal{B}_1, \ldots, \mathcal{B}_{X_m}$. Each bundle $\mathcal{B}_i$ consists of a number of sample paths $S^p(T_m)$. The way in which bundles are determined

is discussed later, but the idea is that sample paths in the same bundle will have values of the basis functions that are "close", i.e.

$$S^{p_1}(T_m), S^{p_2}(T_m) \in \mathcal{B}_i \quad \text{implies} \quad \left|\phi_k(T_{m+1}, S^{p_1}(T_{m+1})) - \phi_k(T_{m+1}, S^{p_2}(T_{m+1}))\right| \quad \text{is "small"}, \tag{16}$$

for all k and any sample paths p_1, p_2, so that when the regression is performed within a bundle then a local approximation to the value function is constructed.

2.2.3 Greeks

In contrast to LSM, the Greeks 'Delta' and 'Gamma' can be obtained by SGBM at almost no extra costs, but the accuracy depends on the quality of the basis functions. For example, the option Delta is defined as

$$\Delta := \frac{\partial V(T_0, S(T_0))}{\partial S(T_0)},$$

and can be approximated (assuming it is not optimal to exercise at T_0) as

$$\begin{aligned} \Delta &= \lim_{x \to 0} \frac{V(T_0, S(T_0) + x) - V(T_0, S(T_0))}{x} \\ &= \lim_{x \to 0} \frac{\sum_{k=1}^{K} \beta_k(T_1)\big(\psi_k(T_0, S(T_0) + x) - \psi_k(T_0, S(T_0))\big)}{x} \\ &= \sum_{k=1}^{K} \beta_k(T_1) \frac{\partial \psi_k(S(T_0))}{\partial S(T_0)}. \end{aligned} \tag{17}$$

Note that the $\beta_k(T_1)$ are the parameters determined during the iteration (backward induction) at time T_0. A generalization to multiple underlying assets is straight forward, and described in [6].

2.3 Low Biased Price Estimate

With both LSM or SGBM it is easy to construct a low biased path estimator once an exercise policy has been obtained. We generate a new set of sample paths and apply the previously found exercise boundary to them; as soon as a path crosses the boundary, we exercise the option and compute the discounted payoff.

This calculation is cheap as no additional regressions are required, however, numerical results show that a large number of paths are required to reduce the variance of the final price approximation suitably.

3 Bermudan Swaption on the LIBOR Market Model (LMM)

We briefly define the LIBOR Market Model just to fix notation, and then derive a pricing formula for Bermudan swaptions driven by it.

3.1 The LIBOR Market Model

Consider a finite set of *tenor* dates $\{T_m\}_{m=0}^{M+1}$, such that $0 = T_0 < T_1 < \ldots < T_{M+1}$ and denote by δ_m the length of the period $[T_m, T_{m+1}]$. Denote the *forward LIBOR rate* over $[T_i, T_{i+1}]$ as of time $0 \le t \le T_i$ by $L_i(t)$ and write the vector of forward rates as the $(M+1)$-dimensional vector $\mathbf{L}(t) = (L_0(t), \ldots, L_M(t))$. LIBOR rates are stochastic simple interest rates, which we consider as the risk-free rates. Note that $L_i(t)$ only exists when $t \le T_i$.

We define a bank account $B^*(T_m)$ as the value at time T_m of an investment of one unit of currency at time T_0, which we can write in terms of the LIBOR rates as

$$B^*(T_m) \equiv B^*(\mathbf{L}(T_m)) = \prod_{j=0}^{m-1} [1 + \delta_j L_j(T_j)], \qquad \text{for } m = 1, \ldots, M+1, \tag{18}$$

and which we will take as the numeraire asset. The associated measure is called the *spot measure*.

Denote by $P_i(t)$ the time t value of a zero-coupon bond, paying one unit of currency at maturity T_i, for $i = 0, \ldots, M+1$, where $P_i(t)$ is defined for $0 \le t \le T_i$. Then we can express the forward LIBOR rates as

$$L_i(t) = \frac{P_i(t) - P_{i+1}(t)}{\delta_i P_{i+1}(t)}, \quad 0 \le t \le T_i, \quad i = 0, 1, \ldots, M. \tag{19}$$

We can invert this equality for a tenor date T_m, and write bond value $P_i(T_m)$ as

$$P_i(T_m) = \prod_{j=m}^{i-1} [1 + \delta_j L_j(T_m)]^{-1}, \qquad \text{for } i = m, \ldots, M+1. \tag{20}$$

Note that it is not possible in LMM to compute $P_i(t)$ for an arbitrary $0 < t \le T_i$, as LIBOR rates do not specify accrual over periods shorter than the difference between tenor dates.

We model the forward LIBOR rates by a system of SDEs of the form

$$\frac{dL_i(t)}{L_i(t)} = \mu_i(t, \mathbf{L}(t))dt + \mathbf{s}_i(t)^\top d\mathbf{W}(t), \quad 0 \le t \le T_i, \text{ and } i = 1, \ldots, M, \tag{21}$$

where $\mathbf{W}$ is a d-dimensional standard Brownian motion under the spot measure, $\mathbf{s}_i$ is a d-dimensional vector and $\mathbf{s}_i^\top$ denotes its transpose. We allow μ_i and $\mathbf{s}_i$ to depend on the current vector of rates $\mathbf{L}(t)$ and the current time t, but we assume $\mathbf{s}_i$ to be deterministic. Using the restriction that the discounted bond prices have to be martingales under this measure we find that the drift parameter must be given by

$$\mu_i(t, \mathbf{L}(t)) = \sum_{j=\eta(t)}^{i} \frac{\delta_j L_j(t) \mathbf{s}_i(t)^\top \mathbf{s}_j(t)}{1 + \delta_j L_j(t)}, \tag{22}$$

where $\eta(t)$ is the unique integer satisfying $T_{\eta(t)-1} \le t < T_{\eta(t)}$.

Since a swaption is an option on an interest rate swap, we briefly examine the basics of swaps first.

3.2 Interest Rate Swap

For any times $T_m \le T_r$ define the forward swap value $S_{[r]}(T_m, \mathbf{L}(T_m))$ to be the value at time T_m and state $\mathbf{L}(T_m)$ of a receiver interest rate swap over the time interval $[T_r, T_M]$ with payment dates $T_{r+1}, \ldots, T_{M+1}$. The holder of the swap receives a fixed interest rate K and pays the floating LIBOR rate over some notional (face) value F. The swap value is found by summing over all its payments: from (19) we have

$$S_{[r]}(T_m, \mathbf{L}(T_m)) = \sum_{j=r}^{M} F\, \delta_j P_{j+1}(T_m) \Big(K - L_j(T_m) \Big), \tag{23}$$

for $m \leq r$. The future swap rate $R_{[r]}(T_m, \mathbf{L}(T_m))$ for $m \leq r$ is the rate K that makes the swap $S_{[r]}(T_m, \mathbf{L}(T_m))$ fair, in other words that solves $S_{[r]}(T_m, \mathbf{L}(T_m)) = 0$ for K. This is just

$$R_{[r]}(T_m, \mathbf{L}(T_m)) = \sum_{i=r}^{M} \left[\frac{P_{i+1}(T_m)}{\sum_{j=r}^{M} P_{j+1}(T_m)} \right] L_i(T_m), \tag{24}$$

thus the swap rate can be written as a linear combination of forward LIBOR rates with stochastic weights. Using (24) we can formulate the value for the receiver swap in terms of the swap rate, which gives

$$S_{[r]}(T_m, \mathbf{L}(T_m)) = \Big(K - R_{[r]}(T_m, \mathbf{L}(T_m)) \Big) \sum_{j=r}^{M} F\, \delta_j P_{j+1}(T_m). \tag{25}$$

In contrast to a receiver swap, a payer swap is an interest rate swap in which the holder of the swap pays the fixed rate. The value of a payer swap is equal to the value of the receiver swap, but of opposite sign.

3.3 Bermudan Swaption

A swaption is an option on a swap. A Bermudan swaption allows multiple times at which the holder can exercise the option and enter into the underlying swap. With T_M the maturity of the option, we let T_r be the *reset* date with $0 < r \leq M$. The holder of the option may exercise it at any tenor between T_r and T_M, but may not exercise it before time T_r. To value a Bermudan swaption, we need to find the optimal stopping time τ taking values in $\mathcal{T} := \{r, r+1, \ldots, M\}$ which maximises the option value

$$V(T_0) = \max_{\tau \in \mathcal{T}} \mathbb{E} \left[\frac{S_{[\tau]}(T_\tau, \mathbf{L}(T_\tau))}{B^*(T_\tau)} \right], \tag{26}$$

where $\mathbb{E}$ is the expectation under the spot-measure corresponding to the numeraire B^*. We denote the set of simulated LMM sample paths by $\{\mathbf{L}^p(T_m) : m = 1, \ldots, M; p = 1, \ldots, N\}$. The payoff function $h(\mathbf{L}(T_m))$ of a Bermudan swaption is given by

$$h(\mathbf{L}(T_m)) = \begin{cases} \max\left(S_{[m]}(T_m, \mathbf{L}(T_m)), 0 \right), & \text{if } r \leq m \leq M, \text{(exercise dates)} \\ 0, & \text{otherwise.} \end{cases} \tag{27}$$

Pricing proceeds by backward induction. For the final time step T_M the swaption value is simply equal to its payoff

$$V(T_M, \mathbf{L}^p(T_M)) = h(\mathbf{L}^p(T_M)). \tag{28}$$

At each preceding time step $T_{M-1}, \ldots, T_1$ the option value is given by (4) for which we need to compute a continuation value. For both LSM and SGBM this requires a set of basis functions.

3.4 Basis Functions

The choice of basis functions under LSM and SGBM mainly relies on experience and numerical tests. It was shown in [7] that the number of sample paths needed grows exponentially with the number of basis functions used. Over specification resulting from too many basis functions makes the regression more sensitive to outliers. Therefore we look for just a few basis functions that are highly correlated to the Bermudan swaption value.

Using the full set of zero coupon bonds or forward rates $\mathbf{L}(t)$ as the basis would result in too many basis functions, especially when one bears in mind that swaptions are sensitive

to correlations between LIBOR rates, which means that cross-product terms would also be needed in the basis.

A process that captures almost all the dynamics of the Bermudan swaption is the underlying swap. In [3] the author suggests to use the swap rate of the nearest-to-maturity swap as a basis function, as well as its square (and a constant). For LSM we therefore set $K = 3$ and define

$$\begin{aligned}\phi_k(T_m, \mathbf{L}(T_m)) &= \begin{cases} R_{[m]}(T_m, \mathbf{L}(T_m))^{k-1} & \text{if } r \leq m \leq M, \\ R_{[r]}(T_m, \mathbf{L}(T_m))^{k-1} & \text{if } m < r \end{cases} \\ &= R_{[\max(m,r)]}(T_m, \mathbf{L}(T_m))^{k-1},\end{aligned} \tag{29}$$

for $k = 1, 2, 3$. For SGBM we will see that fewer basis functions are needed.

3.5 Simulation of the LIBOR Market Model

In (21) the term $\mathbf{s}_i(t)$ determines both the correlation and volatility structure. It is a common choice to write

$$\mathbf{s}_i(t) = \sigma_i(t)\boldsymbol{\lambda}_i(t), \tag{30}$$

where $\sigma_i(t)$ determines the volatility of the forward LIBOR rate $L_i(t)$ and $\boldsymbol{\lambda}_i(t)$ is a vector specifying the correlation among forward rates. This allows independent calibration of correlation and volatility. We choose a piece-wise constant deterministic volatility function of the parametric form

$$\sigma_i(t) = v_i \exp\{-\gamma(T_i - T_{\eta(t)})\}, \tag{31}$$

where v_i is an individual factor for the process $L_i(t)$, and γ is a fixed common component. For a fixed γ, the v_i are calibrated to caplet implied volatilities. Then γ is chosen such that our system calibrates to the quoted price for a European swaption in [2].

To simulate the SDEs for the LIBOR rates we apply an Euler scheme to $\log(L_i(T_m))$. This is equivalent to approximating L_i by a geometric Brownian motion over $[T_m, T_{m+1}]$ with drift and volatility parameters frozen at T_m. This method is attractive since for a deterministic volatility, L_i is close to log-normal. The Euler approximation is then given by

$$L_i(T_{m+1}) \approx L_i(T_m) \exp\left(\left[\mu_i(T_m, \mathbf{L}(T_m)) - \frac{1}{2}\sigma_i(T_m)^2\right]\delta_i + \sqrt{\delta_i}\sigma_i(T_m)\mathbf{Z}_{m+1}\right), \tag{32}$$

where $\mathbf{Z}_{m+1}$ is a d-dimensional vector of correlated normal random numbers where the correlation is specified by $\boldsymbol{\lambda}_i(T_m)$. The drift $\mu_i(T_m, \mathbf{L}(T_m))$ is as in (22), but we emphasize that in the drift the LIBOR rates $\mathbf{L}(T_m)$ are taken to be constant over the interval $[T_m, T_{m+1}]$. Note also that for Euler time steps we use the tenor dates $T_1, \ldots, T_M$ and do not simulate the process at times between tenor dates, which is justified since the volatility function $\sigma_i(t)$ is piecewice constant between tenor dates.

4 Pricing Bermudan Swaptions with SGBM

Two difficulties arise when pricing Bermudan swaptions under the LMM. The first is due to the fact that we are working with a high-dimensional system of equations, and bundling a high-dimensional space is expensive. The second problem is the choice of basis functions since an analytic expression (or suitably accurate approximation) is required for the conditional expectations ψ_k.

4.1 Conditional Expectations of Basis Functions

Ideally one would use the same basis functions (29) for SGBM as one does for LSM. However (recall (14) and (15)) SGBM requires a functional form for the conditional expectation

$$\psi_k(T_m, \mathbf{L}(T_m)) = \mathbb{E}\left[\frac{B^*(T_m)}{B^*(T_{m+1})} \phi_k(T_{m+1}, \mathbf{L}(T_{m+1})) \;\middle|\; \mathbf{L}(T_m) \right], \tag{33}$$

for each k where the expectation is taken under the spot-measure. The availability of such analytic expressions is one of the biggest factors affecting the choice of basis.

For $\phi_1 \equiv 1$ we find from (18) that

$$\psi_1(T_m, \mathbf{L}(T_m)) = \mathbb{E}\left[\frac{B^*(T_m)}{B^*(T_{m+1})} \;\middle|\; \mathbf{L}(T_m) \right] = \frac{1}{1 + \delta_m L_m(T_m)}. \tag{34}$$

For $\phi_2(T_m, \mathbf{L}(T_m)) = R_{[\max(m,r)]}(T_m, \mathbf{L}(T_m))$, however, it is easily shown via Ito's Lemma that the swap rate is not log-normal. Thus, the conditional expectation is difficult to compute. Luckily, the swap rate is close to log-normal as shown in [3]. Using an approximation introduced by Rebonato (see e.g. [2]), we can approximate the swap rate by a log-normal process.

Recall that the underlying swap rate of the nearest-to-maturity swap as of T_{m+1} is given by

$$R_{[\max(r,m+1)]}(T_{m+1}, \mathbf{L}(T_{m+1})) = \sum_{i=\max(r,m+1)}^{M} \left[\frac{P_{i+1}(T_{m+1})}{\sum_{j=\max(r,m+1)}^{M} P_{j+1}(T_{m+1})} \right] L_i(T_{m+1}),$$

in other words a weighted average of LIBOR rates. The weight factors are stochastic, but [3] argues that their volatility is small compared to the volatility of the forward LIBOR rates. If we approximate the weights by their known values at time T_m, then we find that the swap rate is a linear combination of forward rates at time T_{m+1}. The conditional expectation of these forward rates can be readily approximated as well if we use (32) to approximate $L_i(T_{m+1})$ by a log-normal variable

$$\mathbb{E}\left[\frac{B^*(T_m)}{B^*(T_{m+1})} L_i(T_{m+1}) \;\middle|\; \mathbf{L}(T_m) \right] \approx \frac{L_i(T_m)}{1 + \delta_i L_i(T_m)} e^{\mu_i(T_m, \mathbf{L}(T_m))\delta_i}. \tag{35}$$

Since the forward swap rate is approximated by a linear combination of the forward rates, we find an approximation for the conditional expectation

$$\psi_2(T_m, \mathbf{L}(T_m)) \approx \sum_{i=\max(r,m+1)}^{M} \left[\frac{P_{i+1}(T_m)}{\sum_{j=\max(r,m+1)}^{M} P_{j+1}(T_m)} \right] \frac{L_i(T_m) e^{\mu_i(T_m, \mathbf{L}(T_m))\delta_i}}{1 + \delta_m L_m(T_m)}. \tag{36}$$

Thus the swap rate can be used as a basis function under SGBM provided the approximation (36) is accurate enough.

In [6], SGBM was applied to several options driven by geometric Brownian motion. It was shown that in this model a constant and a first order basis function served well enough to approximate the option price. By contrast, LSM requires at least second order bases as well. We note that using the squared swap rate as a basis function in SGBM would be difficult since approximation formulas for the squared rate are not readily available. For Greeks, especially Gamma, higher order basis functions are typically required.

4.2 Bundling the LIBOR Market Model

The LMM contains $M+1$ different underlying stochastic forward rates. Recall from Section 2.2.2 that the purpose of bundling is to partition the state space into small pieces so that local approximations can be made on each piece.

Partitioning the high dimensional state space of a non-trivial process in this way leads to exponential growth in the number of bundles, and hence exponential growth in the computational complexity of the SGBM algorithm. In addition, a huge number of sample paths would be needed to ensure a sufficient number in each bundle. High dimensional spaces therefore have to be handled by some form of dimension reduction. Typically, one chooses a mapping function f which takes the state $\mathbf{L}(T_m)$ and maps it down to a target space, ideally $\mathbb{R}$. The bundling can then be performed in $\mathbb{R}$ on the mapped values. A good mapping function should have the following properties:

1. when the target space is bundled, each bundle should contain sample paths which have "similar" values of the basis functions. This addresses (16) and ensures that when we regress paths within a bundle, we do indeed construct a local approximation to the option value

2. the mapped value should ideally be a sufficient statistic for the computation of the conditional expectations of the basis functions (the ψ_k's). This would avoid having to invert the mapping function since by assumption given $f(T_m, \mathbf{L}(T_m))$ we would be able to compute each $\psi_k(T_m, \mathbf{L}(T_m))$ without needing the value of $\mathbf{L}(T_m)$. This is not necessary but it is convenient.

Since we have only one non-trivial basis function, namely the swap rate, we can take as our mapping function ψ_2, the conditional expectation of the swap rate. Then the second condition above is trivially satisfied, while the first is true in an L_2 sense since the conditional expectation ψ_2 is the best prediction of ϕ_2 in the L_2 metric.

To determine the bundles in $\mathbb{R}$ we apply a so-called *equal number bundling* on X_m bundles at time T_m. The aim is to get roughly an equal number of sample paths in each bundle. This method of bundling was selected primarily for its speed and simplicity. In [5] the authors argue that the quality of the bundling is not very important when good basis functions are used.

To speed up the bundling we use only the set of $N_{train} \ll N$ paths to generate bundles. At each time step we order the training paths with respect to the mapping function, and paths with rankings between $\frac{(j-1)N}{X_m} + 1$ and $\frac{jN}{X_m}$ belong to the j^{th} bundle. The minimum and maximum path values in each bundle are then taken to be the bundle boundaries.

Once the bundles at each time step have been fixed, we generate the full set of N sample paths and use the bundle boundaries to assign paths to bundles.

5 Numerical Results

We present numerical results comparing the pricing of Bermudan swaptions on the LIBOR Market Model with SGBM and LSM. We consider a constant time $\delta = \frac{1}{2}$ between tenor dates, and set $T_m = m\delta$, for $m = 0, \ldots, 20$. We value an ATM Bermudan swaption with reset date $T_r = T_8 = 4$ years and maturity $T_M = T_{20} = 10$ years, having semi-annual payments and exercise moments T_j, for $j = 8, \ldots, 20$.

To test the accuracy and computational time of SGBM against LSM, we calibrate a one factor LMM according to the initial LIBOR curve shown in Table 1 with the caplet implied volatilities shown in Table 2 (the data comes from [2]). The volatility decay parameter γ in (31) is set to 0.07.

In SGBM, three types of errors can occur when approximating the option value:

T_i	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5
$L(0,T_i)$	422.8	279.1	306.7	306.7	372.8	372.8	405.1	405.1	419.9	419.9
T_i	5	5.5	6	6.5	7	7.5	8	8.5	9.0	9.5
$L(0,T_i)$	445.0	445.0	462.6	462.6	481.6	481.6	496.0	496.0	508.8	508.8

Table 1: Forward LIBOR curve (in BPS) on 18 November 2008. Source: [2]

i	1	2	3	4	5	6	7	8	9	10
$\sigma_{capl(T_{i-1},T_i)}$	n/a	29.3	29. 3	29.3	20.8	20.8	18.3	18.3	17.8	17.8
i	11	12	13	14	15	16	17	18	19	20
$\sigma_{capl(T_{i-1},T_i)}$	16.3	16.3	16.7	16.7	16.1	16.1	15.7	15.7	15.7	15.7

Table 2: Implied volatilities $\sigma_{capl(T_{i-1},T_i)}$ (in %) for caplets over $[T_{i-1}, T_i]$ at strike rate 3.5%. Source: [2]

E1 The error in approximating the option value by a linear combination of basis functions. In [1] it is shown that this error can be reduced efficiently by bundling.

E2 The error in approximating the conditional expectation of the swap rate $\psi_2(T_m, \mathbf{L}(T_m))$ by the formula (36).

E3 The error in estimating the regression parameters based on the sampled data, which is related to the number of paths.

According to the Central Limit Theorem, E3 can be reduced by increasing the number of sample paths. We study the other errors numerically.

5.1 Swap Rate Approximation (SRA) Error

The error E2 is caused by using the SRA formula (36) to compute $\psi_2(T_m, \mathbf{L}(T_m)) \approx \mathbb{E}[\,\phi_2(T_{m+1}, \mathbf{L}(T_{m+1}) \mid \mathbf{L}(T_m)]$. This approximation consists of two parts. Firstly, the 'weights are frozen', i.e., the bond prices in (36) are frozen at T_m. Secondly, the forward rates $L_i(T_{m+1})$ are approximated by a log-normal random variable which allows us to compute the expectation.

To test the accuracy of these approximations, we ran a nested Monte Carlo simulation to estimate the conditional expectations for a variety of initial LIBOR curves. Since the discretization scheme we use also approximates the forward rates as log-normal random variables, both the Monte Carlo estimate and the approximation suffer from this discretization bias. We take the nested Monte Carlo result as the true value and therefore the error in the SRA will be dominated by 'freezing the weights'. The variation in the weights is far less than the variation in the forward rates and it takes over 10^9 sample paths to reduce the variance of the Monte Carlo estimate, enough to make the error in the approximation visible.

Figure 1 shows the results of the nested Monte Carlo simulation. We simulated 100 different LIBOR curves at the reset date T_8 from the single factor LIBOR model described above. At this date, the swap rate still depends on all the forward rates $L_i(T_8)$, $i = 8, \ldots, M$, as can be seen in (36), and thus the worst approximation can be expected at this time. For each of the 100 forward curves generated, we used an inner Monte Carlo

simulation of 10^9 paths to estimate the conditional expectation. This value (denoted MC in the figure) was compared to the swap rate approximation formula SRA.

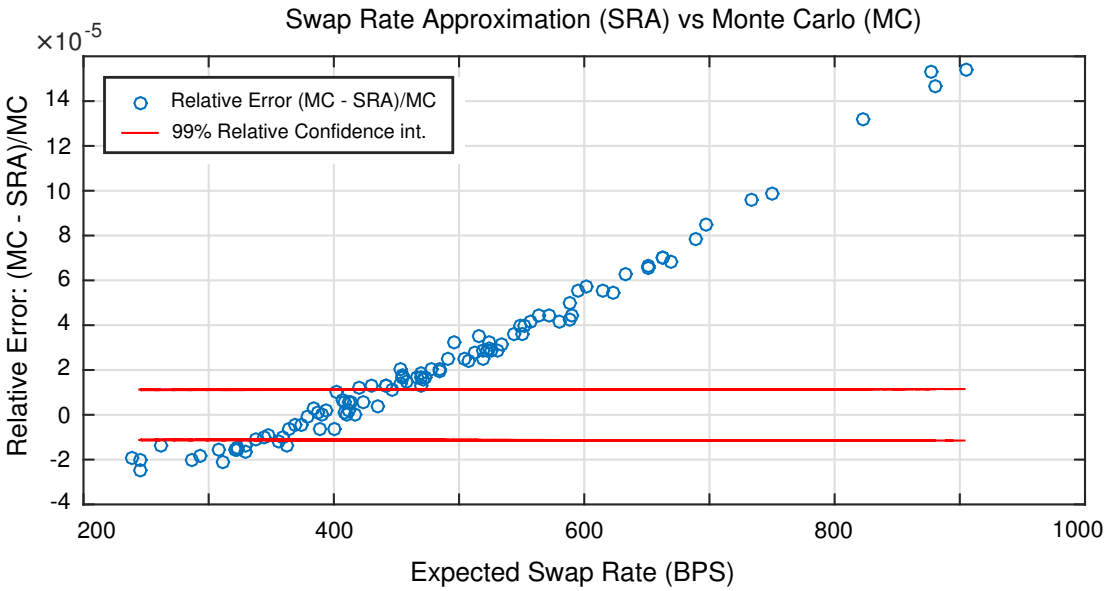

Figure 1: SRA formula vs MC for 100 simulated forward curves in the one-factor LMM. The solid line plots $3\frac{\sigma}{\sqrt{10^9}}/\text{MC}$ where σ is the standard error of the MC estimate. The solid line is therefore a 99% confidence interval for $(\text{MC} - \psi_2)/\text{MC}$. It is clear that the error in the SRA increases as the future swap rate increases.

On average, the approximation formula for ψ_2 tends to underestimate the true expected value, and the relative error increases linearly when the expected swap rate increases. When pricing a receiver swaption, the option value is negatively correlated to ψ_2, which implies that SGBM overestimates the option price, and the opposite is true for payer swaptions.

5.2 Effects of Bundling

In both SGBM and LSM there are two phases to the calculation:

- In the first phase (also called the *direct estimate*) the optimal exercise strategy is estimated. This procedure is based on Monte Carlo simulation together with regression onto basis functions. This phase also produces an option price estimate
- The second phase is optional: given an exercise strategy, we can generate a new batch of sample paths and apply the strategy to them to get a lower bound for the true option price. This phase is called the *path estimator*.

The first phase is by far the most expensive in terms of runtime, whereas the second phase is almost identical for both methods. We consider the SGBM with $4, 8$ and 16 bundles. We did not use more bundles in order to avoid over-fitting the option value surface. A large number of bundles will give a price estimate with a low standard error, but a significant bias can occur.

Since SGBM generates a single price estimate, an estimate of the standard error of the price is not readily available. We therefore computed 10 different SGBM price estimates and took the mean and standard error of this sample.

We see that by increasing the number of bundles the price estimates for SGBM converges, and the exercise strategy of the lower-bound SGBM path estimator outperforms the LSM path estimator for both receiver and payer swaptions. It was shown in [6] that

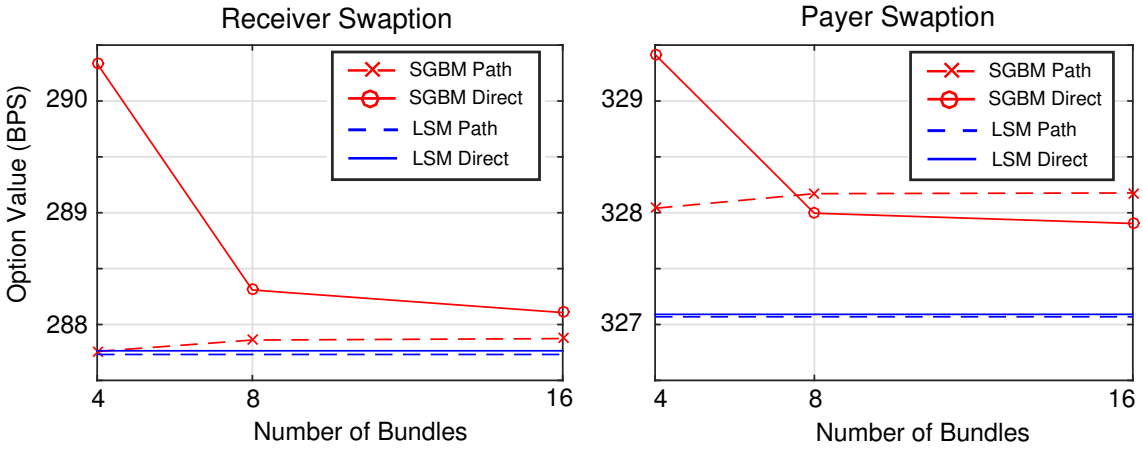

Figure 2: Comparison of SGBM and LSM for the receiver and payer swaption. We simulated 10 batches of $N = 10^6$ sample paths. Standard errors are too small to plot.

when the conditional expectation of the basis functions are exact, the SGBM direct estimator is high-biased. This is what what we observe for the receiver swaption (indeed the situation is made worse due to the low bias of the SRA and the negative correlation, see (25), between the option value and the basis function). For the payer swaption the SGBM direct estimate is however not high-biased due to the low bias in the SRA and a positive correlation between the option value and the basis function. It is interesting that the low-biased path estimator gives a better (higher) price than the direct estimator itself.

We found that SGBM with 8 bundles is sufficient to effectively reduce bias (see Figure 2), and is small enough not to suffer from over-fitting when using only 10^3 paths.

5.3 Computational Time

We compare the computational time of the first phase (direct estimator) of SGBM against LSM in terms of the standard errors of the price estimates.[a] Numerical experiments show that the computational time for SGBM is practically independent of the number of bundles, thus we limit the discussion to SGBM with 8 bundles. For the same number of LIBOR sample paths, the computational time for the SGBM direct estimator is around six times larger[b] than the LSM direct estimator due to the additional computation of the ψ_k's. One advantage of SGBM is that regressions are split up per bundle, thus each regression is smaller and regressions can be performed in parallel, but a synchronization between parallel threads is still required every time step since paths can change bundle over time.

However, as is clear from Figure 3 the standard error of the SGBM direct estimator is much lower than the standard error of the LSM direct estimator. This is a big advantage: it means that fewer sample paths can be used in the SGBM direct estimator than in the LSM direct estimator and path generation under LMM is typically expensive. SGBM requires about 10 times fewer paths than LSM to get a price estimate with a similar standard error. Figure 4 compares the standard error and overall runtime (including LIBOR path generation) for the LSM and SGBM direct estimators. All in all, we find that SGBM is around 6 times faster than LSM for a given level of accuracy.

Turning to the path estimators, we find (as expected) that the SGBM and LSM estimators have the same standard error. The computational time for the SGBM path estimator

[a] We performed the experiments on an 2.66 GHz Intel Core 2 duo with 8 GB of memory and the algorithms are implemented in `C`. Regression and random number generation are performed using the NAG Library. Parallelization of the path generation (over two threads) is performed with OpenMP.

[b] LSM direct estimator vs SGBM direct estimator is 0.08s vs 0.6s at $N = 10,000$; 1.0s vs 6.0s at $N = 100,000$; and 8.0s vs 50.0s at $N = 1,000,000$

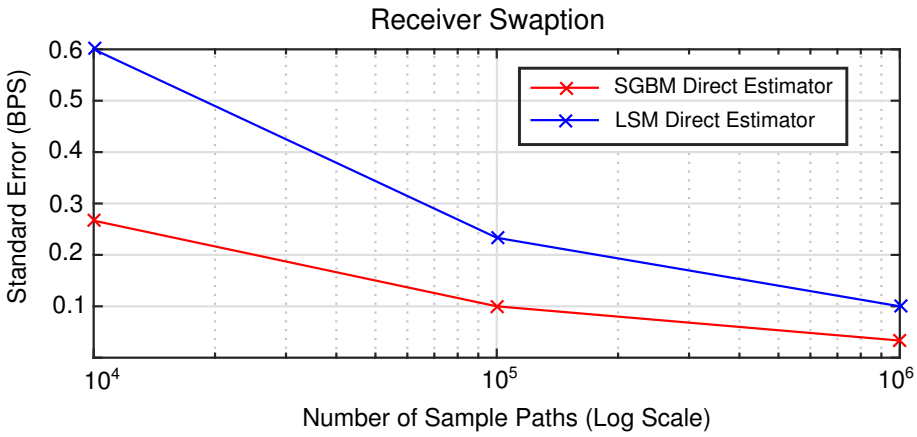

Figure 3: Standard errors of direct estimators vs number of sample paths for the receiver swaption. The figure for payer swaption is similar. Note that SGBM requires around 10 times fewer sample paths to achieve the same standard error as LSM.

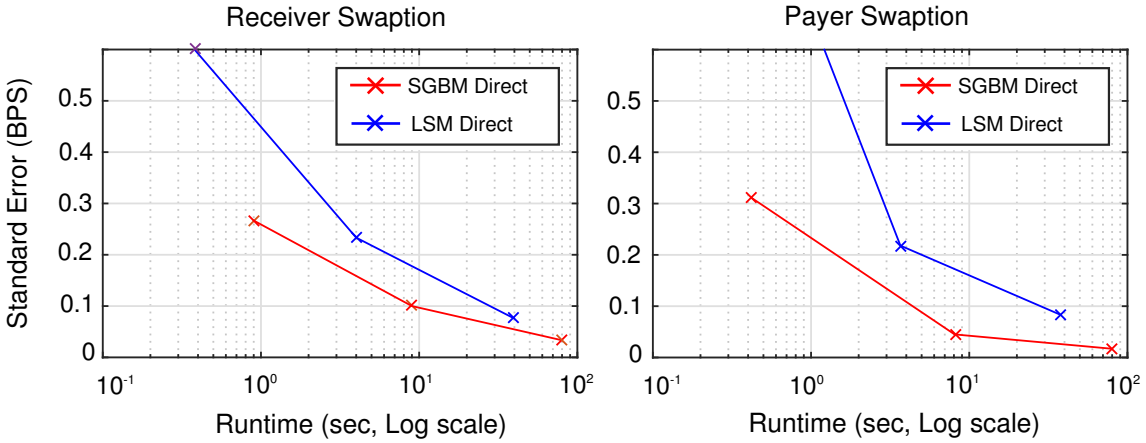

Figure 4: Overall runtime (LIBOR path generation + direct estimator) vs standard error for receiver and payer swaptions. Note that SGBM is around 6 times faster than LSM for the same level of standard error.

is about five times[c] more than the LSM path estimator due to the additional computation of the ψ_k's. With careful coding it should be possible to reduce this overhead. The path estimator runtimes are much lower than the direct estimator runtimes.

6 Conclusions and Outlook

In this paper we explored how to apply the novel Stochastic Grid Bundling Method to a high-dimensional problem, the Bermudan swaption on the LIBOR Market Model, and compared it to Longstaff–Schwartz. The two main challenges with SGBM are to choose a proper set of basis functions and a bundling strategy. For the basis functions we used a constant and the nearest-to-maturity swap rate. The conditional expectation of the swap rate was computed via a lognormal approximation formula, and the bundling was performed on this conditional expectation.

[c]LSM path estimator vs SGBM path estimator is 0.02s vs 0.1s at $N = 10,000$; 0.2s vs 1.0s at $N = 100,000$; and 2.0s vs 10.0s at $N = 1,000,000$

With the swap rate approximation formula, the SGBM price agrees with the LSM price up to a basis point, and the SGBM path estimators give better (higher) prices than the LSM prices. The current approach cannot in general deliver highly accurate option prices due to the inaccuracies in the approximation formula, and to an inherent bias in the SGBM method. The main advantage of SGBM is that it requires fewer paths to obtain a price estimate with a similar standard error as LSM, which results in SGBM being around 6 times faster than LSM overall.

The SGBM can therefore serve as a fast method to get reasonably accurate prices.

Further improvements can be made by improving the swap rate approximation and exploring different basis functions or bundle mappings. Also, more extensive testing should be done under different calibrations and volatility structures of the driving LMM.

References

[1] Feng, Q. and Oosterlee, C.W. (2013) Monte Carlo Calculation of Exposure Profiles and Greeks for Bermudan and Barrier Options under the Heston Hull-White model. *papers, SSRN*.

[2] Filipović, D. (2009) Term-Structure Models, A Gradate Course. *Springer*.

[3] Glasserman, P. (2003) Monte Carlo Methods in Financial Engineering. *Springer*.

[4] Glasserman, P. and Yu, B. (2004) Simulation for American options: Regression Now or Regression Later? *Springer verlag*.

[5] Jain, S. and Oosterlee, C.W. (2012) Pricing high-dimensional Bermudan options using the stochastic grid method. *International Journal of Computer Mathematics*, 89(9):1186-1211.

[6] Jain, S. and Oosterlee, C.W. (2013) The Stochastic Grid Bundling Method: Efficient Pricing of Bermudan Options and their Greeks. *papers, SSRN*.

[7] Leclerc, M., Liang, Q. and Schneider, I. (2009) Fast Monte Carlo Bermudan Greeks. *Risk Magazine*.

[8] Longstaff, F. and Schwartz, E. (2001) Valuing American Options by Simulation: a Simple Least-squares Approach. *Review of Financial Studies*, 14(1):113-147.

7 List of NAG C Library Routines Used

The following NAG C Library routines were used to implement the Stochastic Grid Bundling Method and Longstaff–Schwartz Method on a one factor LIBOR Market Model:

```
nag_init_repeatable, nag_rand_normal, nag_summary_stats_1var,
nag_double_quantiles, nag_regsn_mult_linear
```