To appear in *XXXXX Finance*, Vol. 00, No. 00, Month 20XX, 1–29

Rolling Adjoints : fast Greeks along Monte Carlo scenarios for early-exercise options

Shashi Jain * † and Alvaro Leitao‡ and Cornelis W. Oosterlee§

†Department of Management Studies, Indian Institute of Science, Bangalore, India
‡Riskcenter Group, University of Barcelona, Barcelona, Spain
§ Centrum Wiskunde and Informatica, Amsterdam, the Netherlands.

(*v1.0 released December 2017*)

In this paper we extend the stochastic grid bundling method (SGBM), a *regress-later* based Monte Carlo scheme for pricing early-exercise options, with an *adjoint* method to compute in a highly efficient manner sensitivities along the paths, with reasonable accuracy. With the ISDA standard initial margin model being adopted by the financial markets, computing sensitivities along scenarios is required to compute quantities like the margin valuation adjustment.

Keywords: Greeks, Monte Carlo, MVA, sensitivities along paths, American Monte Carlo

JEL Classification: Please provide at least one JEL Classification code

1. Introduction

The efficient calculation of financial instrument sensitivities by Monte Carlo methods is a problem of practical importance. The ISDA standard initial margin model (SIMM) for non-standard derivatives, such as callable exotic options, is based on sensitivities of financial derivatives to determine today's initial margins. To compute the margin valuation adjustment (MVA) to manage the funding costs for posting initial margin over the lifetime of a derivative, sensitivities along the paths need to be computed. Computing sensitivities along the paths, for all the underlying risk factors, for all non-standard derivatives, for each monitoring date[1], will pose significant computational challenges. Additionally, if the non-standard derivative is defined by embedded early-exercise features, computing sensitivities along the paths poses additional challenges.

The simplest approach for estimating a price sensitivity with respect to a parameter is based on a finite difference approximation that is commonly known as the *bump-and-revalue* (*BR*) approach. It is based on performing the Monte Carlo technique multiple times with respect to shifted (i.e., bumped) up and down values of the concerned parameters. The wide acceptance of the method is

*Corresponding author. Email: shashijain@iisc.ac.in

[1] we use the term *monitoring date* for the subset of forward dates on which the corresponding option values and their sensitivities need to be computed along the scenarios, for instance, for the purpose of exposure calculation or future SIMM based initial margin estimations

Electronic copy available at: https://ssrn.com/abstract=3093846

attributed to the ease of implementation and its suitability for parallel computing. The drawbacks however are that the bias and variance properties of finite difference estimates can be poor, the computational costs may grow significantly with the number of input parameters for which the sensitivity is required, and the scheme is not suitable for computing sensitivities along the scenarios, that are required for instance to calculate the MVA.

Advanced methods for estimating sensitivities include the path-wise [1] and likelihood ratio methods, as described in Chapter 7 of Glasserman (2003)(13). The path-wise method employs differentiation of the payoff and the underlying assets or state variables along each path. The method requires the payoff to be smooth, although there are techniques to handle non-smooth payoffs such as for digital or barrier instruments. The likelihood ratio method differentiates the transition density of the underlying assets or state variables. In comparison to *BR*, the path-wise and likelihood methods produce better quality sensitivity estimates, although their implementation is often non-trivial and depends on the underlying process and payoffs.

Path-wise sensitivities can be computed either in an ordinary manner, i.e., forward in time, or by adjoints, i.e., backwards in time. The forward and adjoint methods provide the same sensitivity estimates, however, the difference between the two is mainly in the computational effort. As pointed out in the original *Smoking Adjoints* by Giles and Glasserman (2006)(12) the adjoint method is advantageous for calculating the sensitivities of a small number of securities with respect to a large number of parameters, while the forward method is advantageous for calculating sensitivities of many securities with respect to a small number of parameters.

Adjoint differentiation (AD) is technically non-trivial to implement in an existing library and is often achieved by introducing automatic differentiation (aka algorithmic differentiation), where differentiation is performed using derivatives defined at the operator level. It has been shown that the computational complexity of AD is no more than four times the complexity of the original algorithm. The adjoint is calculated after the pricing computations, which implies that information generated during the pricing should be stored for the calculation of the adjoint. The terminology for the structure in which the relevant pricing operations are recorded is known as the tape, which is played back to perform the differentiation using the recorded information. The tape often leads to memory issues and complexity in coding. It should be noted that computing sensitivities along scenarios using the path-wise approach is also non-trivial.

Valuation of financial options with early-exercise features by Monte Carlo methods often involve regression techniques. The least squares Monte Carlo method (LSMC), proposed by Longstaff and Schwartz (2001)(24) is one of the most widely used methods for pricing such options. Briefly, the method approximates recursively, moving backwards in time, the continuation value (a conditional expectation) function by regressing the future optimal discounted cashflows against a set of basis functions of the underlying state variables at that time epoch.

Some other Monte Carlo based schemes for pricing American options are the stochastic mesh method proposed by Broadie and Glasserman (2004)(5), where the involved conditional expectations are approximated, recursively, moving backwards in time, as weighted sums of the future option values at discrete mesh points. In Glasserman and Yu (2004)(14) a variant of LSMC is proposed, where the basis for regression is constructed using the state variables at the next time step, as opposed to the current time step. This approach has commonly been referred to as *regression later* (or, regress-later), or also, as in Beutner *et al.* (2015)(3), as replicating portfolio. An extensive comparative analysis of the regress-now (the usual LSMC) versus portfolio replication scheme can be found in Beutner *et al.* (2013)(4) and Pelsser and Schweizer (2016)(25).

[1] The terminology *path-wise* is used here for the approach where the option sensitivity at time t_0 is calculated using the expectation of the path-wise sensitivities at the expiration date. This should not be confused with sensitivities along the path or scenario, a terminology we use here for the conditional option sensitivities on the monitoring dates along the simulated paths.

Electronic copy available at: https://ssrn.com/abstract=3093846

Application of AD for regress-now Monte Carlo methods has been proposed, amongst others, by Antonov (2017)(1) and Caprioti (2016)(6). In Antonov *et al.* (2015)(2) a scheme called *backward differentiation* is proposed, which circumvents the use of tapes for AD methods.

In Wang and Caflisch (2010)(27) an extension of LSMC is proposed to estimate Delta and Gamma based on an initial state dispersion scheme. It is shown by Letourneau and Stentoft (2016)(23) that for the above method a bias, which is not only a function of the order of the polynomial but is also related to the size of the initial dispersion, is introduced, and it is not possible to propose a method for the initial dispersion which is generally applicable using this methodology. In Letourneau and Stentoft (2016)(23), more structure is brought into the initial sample dispersion scheme. However, the above methods are not suitable for computing parameter related sensitivities such as the option Vega.

There is a growing literature on sensitivity estimators using Malliavin calculus. This line of work originated in Fournie *et al.* (1999)(11), and includes Gobet and Kohatsu-Higa (2003)(15), Kohatsu-Higa and Montero (2004)(20), and many others. Chen and Glasserman (2007)(20) study the link between Malliavin calculus based approaches to the path-wise and likelihood ratio methods. In Leitao *et al.* (2017)(22) a data-driven approach to infer the unknown characteristic functions is proposed which can then be used to estimate option Greeks.

It was shown in Jain and Oosterlee (2015)(17) that efficient approximations of the t_0 option Delta and Gamma can be obtained using SGBM. However, computing model parameter based sensitivities, and the sensitivities along the paths is nontrivial using the original SGBM. In this paper we extend the SGBM method to efficiently compute model parameter sensitivities and Greeks along the paths, using a combination of differentiation of regression coefficients and adjoint differentiation between monitoring dates. The resulting sensitivities at the end of the monitoring period from adjoint differentiation is implicitly rolled over into the sensitivities of the regression coefficients of the previous monitoring date. For this reason we call the method *Rolling Adjoints*, which facilitates *Smoking Adjoints* (2005)(12) to compute conditional sensitivities along the path for options with early-exercise features.

This paper is structured as follows. In Section 2 we formulate the problem and set the notation; in Section 3 we briefly summarize SGBM. In Section 4 we discuss the extension of SGBM to compute sensitivities along paths. In Section 5 we provide extensive numerical examples. In Section 6 we discuss some of the conclusions.

2. Problem Formulation

This section defines the Bermudan option pricing problem and sets up the notations used in this paper. We assume a complete probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and finite time horizon $[0, T]$. Ω is the set of all possible realizations of the stochastic economy between 0 and T. The information structure in this economy is represented by an augmented filtration $\mathcal{F}_t : t \in [0, T]$, with $\mathcal{F}_t$ the sigma field of distinguishable events at time t, and $\mathbb{P}$ is the risk-neutral probability measure on elements of $\mathcal{F}$. It is assumed that $\mathcal{F}_t$ is generated by W_t, a d-dimensional standard Brownian motion, and the state of economy is represented by an $\mathcal{F}_t$-adapted Markovian process, $\mathbf{X}_t = (X_t^1, \dots, X_t^d) \in \mathbb{R}^d$, which has dependence on model parameters $\theta = \{\theta_1, \dots, \theta_{N_\theta}\}$, where $t \in [t_0 = 0, \dots, t_m, \dots, t_M = T]$. Let $h_t := h(\mathbf{X}_t)$ be an adapted process representing the intrinsic value of the option, i.e. the holder of the option receives $\max(h_t, 0)$, if the option is exercised at time t. With the risk-less savings account process, $B_t = \exp(\int_0^t r_s \, ds)$, where r_t denotes the instantaneous risk-free rate of return, we define

$$D_{t_{m-1}} = \frac{B_{t_{m-1}}}{B_{t_m}}.$$

We consider the special case where r_t is constant. The problem is then to compute

$$\frac{V_{t_0}(\mathbf{X}_{t_0})}{B_{t_0}} = \max_{\tau} \mathbb{E}\left[\frac{h(\mathbf{X}_\tau)}{B_\tau}\right], \tag{1}$$

where τ is a stopping time, taking values in the finite set $\{0, t_1, \ldots, T\}$.

The dynamic programming formulation to solve this optimization problem is then as follows. The value of the option at the terminal time T is equal to the product's pay-off,

$$V_T(\mathbf{X}_T) = \max(h(\mathbf{X}_T), 0). \tag{2}$$

Recursively, moving backwards in time, the following iteration is then solved, given V_{t_m} has already been determined, the continuation or hold value $Q_{t_{m-1}}$ is given by:

$$Q_{t_{m-1}}(\mathbf{X}_{t_{m-1}}) = B_{t_{m-1}} \mathbb{E}\left[\frac{V_{t_m}(\mathbf{X}_{t_m})}{B_{t_m}}\middle|\mathbf{X}_{t_{m-1}}\right]. \tag{3}$$

The Bermudan option value at time t_{m-1} and state $\mathbf{X}_{t_{m-1}}$ is then given by

$$V_{t_{m-1}}(\mathbf{X}_{t_{m-1}}) = \max(h(\mathbf{X}_{t_{m-1}}), Q_{t_{m-1}}(\mathbf{X}_{t_{m-1}})). \tag{4}$$

The sensitivity of interest might be either with respect to $\mathbf{X}_{t_{m-1}}$, i.e., $\frac{\partial V_{t_{m-1}}(\mathbf{X}_{t_{m-1}})}{\partial \mathbf{X_{t_{m-1}}}}$, or with respect to the model parameters θ, i.e. $\frac{\partial V_{t_{m-1}}(\mathbf{X}_{t_{m-1}})}{\partial \theta}$.

Assuming minimal smoothness of the option value function V, the following holds true

$$\frac{\partial}{\partial \theta}\left(\mathbb{E}\left[\frac{V_{t_m}(\mathbf{X}_{t_m})}{B_{t_m}}\middle|\mathbf{X}_{t_{m-1}}\right]\right) = \mathbb{E}\left[\frac{\partial}{\partial \theta}\left(\frac{V_{t_m}(\mathbf{X}_{t_m})}{B_{t_m}}\right)\middle|\mathbf{X}_{t_{m-1}}\right],$$

with the conditions for the interchange as discussed in Glasserman (2003)(13).

In a Monte Carlo simulation, the time evolution of the process $\mathbf{X}$ is approximated using some discretization scheme. We define a general Markov discretization scheme as

$$\mathbf{X}_{t_m} = F_{m-1}(\mathbf{X}_{t_{m-1}}, \mathbf{Z}_{t_{m-1}}, \theta), \tag{5}$$

where $\mathbf{Z}_{t_{m-1}}$ is a d-dimensional standard normal random vector, and F_{m-1} is a transformation from $\mathbb{R}^d$ to $\mathbb{R}^d$.

3. Stochastic grid bundling method

In this section we briefly describe the main aspects of SGBM, since a detailed description and convergence study of the method can be found in (17).

SGBM is based on N independent copies of sample paths, $\{\mathbf{X}_{t_0}, \ldots, \mathbf{X}_{t_M}\}$, of the underlying process obtained using the recursion, $\mathbf{X}_{t_m}(n) = F_{m-1}(\mathbf{X}_{t_{m-1}}(n), \mathbf{Z}_{t_{m-1}}(n), \theta)$, where $n = 1, \ldots, N$ is the index of the path. The method then computes the option value at terminal time as $V_{t_M}(\mathbf{X}_{t_M}) = \max(h(\mathbf{X}_{t_M}), 0)$.

The following SGBM components are employed within each time step, t_m, $m \leq M$, moving backwards in time, starting from t_M.

Bundling The grid points at t_{m-1} are *bundled* into $\mathcal{B}_{t_{m-1}}(1), \ldots, \mathcal{B}_{t_{m-1}}(\nu)$ non-overlapping sets or partitions using, for instance, any of the following schemes:

- *k-means clustering algorithm*,
- *recursive bifurcation*,
- *recursive bifurcation of a reduced state space.*

A mapping $\mathcal{I}^{\beta}_{t_{m-1}} : \mathbb{N}^{[1,N_\beta]} \mapsto \mathbb{N}^{[1,N]}$, is defined which maps ordered indices of paths in a bundle $\mathcal{B}_{t_{m-1}}(\beta)$ to the original path indices, where $N_\beta := |\mathcal{B}_{t_{m-1}}(\beta)|$ is the cardinality of the β-th bundle, $\beta = 1, \ldots, \nu$.

Within each bundle $\mathcal{B}_{t_{m-1}}(\beta)$, $\beta = 1, \ldots, \nu$, the following operations are performed

- ***Regress-later* within each bundle**

  Corresponding to each bundle $\mathcal{B}_{t_{m-1}}(\beta)$, $\beta = 1, \ldots, \nu$, a parameterized value function $\tilde{G} : \mathbb{R}^d \times \mathbb{R}^K \mapsto \mathbb{R}$, which assigns values $\tilde{G}(\mathbf{X}_{t_m}, \alpha^{\beta}_{t_m})$ to states $\mathbf{X}_{t_m}$, is computed. Here $\alpha^{\beta}_{t_m} \in \mathbb{R}^K$ is a vector of free parameters. The objective is to choose, for each t_m and β, a parameter vector $\alpha^{\beta}_{t_m}$ so that

  $$\tilde{G}(\mathbf{X}_{t_m}, \alpha^{\beta}_{t_m}) = V_{t_m}(\mathbf{X}_{t_m}).$$

  Assuming the square-integrability of the option value function, it is well-known that under certain conditions (see (4)) the option value function can be expressed as a linear combination of a countable number of orthonormal basis functions. Therefore, one may choose $\widehat{G} \approx \tilde{G}$ to be:

  $$V_{t_m}(\mathbf{X}_{t_m}) = \widehat{G}(\mathbf{X}_{t_m}, \alpha^{\beta}_{t_m}) = \sum_{k=1}^{\infty} \alpha^{\beta}_{t_m}(k)\phi_k(\mathbf{X}_{t_m}), \tag{6}$$

  although in practice the orthonormal basis is restricted to a finite set, i.e. $k \leq K$, and thus

  $$V_{t_m}(\mathbf{X}_{t_m}) \approx G(\mathbf{X}_{t_m}, \alpha^{\beta}_{t_m}) = \sum_{k=1}^{K} \alpha^{\beta}_{t_m}(k)\phi_k(\mathbf{X}_{t_m}), \tag{7}$$

The economic interpretation of this step is that, conditional on a path ending up in a particular bundle at t_{m-1}, a static portfolio is set up to replicate the option value at t_m. The K basis functions chosen then represent the payoffs of the hedge instruments, and $\alpha^{\beta}_{t_m}$ are the corresponding weights of these instruments in the portfolio. All paths within a bundle would then have the same set of hedge instruments and portfolio weights.

The portfolio weights are approximated using a least squares approach, i.e.,

$$\underset{\widehat{\alpha}^{\beta}_{t_m}}{\operatorname{argmin}} \sum_{n=1}^{N_\beta} \left(V_{t_m}\left(\mathbf{X}_{t_m}\left(\mathcal{I}^{\beta}_{t_{m-1}}(n)\right)\right) - \sum_{k=1}^{K} \widehat{\alpha}^{\beta}_{t_m}(k)\phi_k\left(\mathbf{X}_{t_m}\left(\mathcal{I}^{\beta}_{t_{m-1}}(n)\right)\right)\right)^2. \tag{8}$$

Given a finite number of sample paths used for the least squares regression, the coefficients $\widehat{\alpha}^{\beta}_{t_m}$ represent the approximation for $\alpha^{\beta}_{t_m}$.

In Appendix A we illustrate by means of a basic example the importance of bundling for the *regress-later* scheme, which otherwise can have a significant bias for path-dependent options.

- **Computing the continuation and option values**

  The continuation values for $\mathbf{X}_{t_{m-1}}(n) \in \mathcal{B}_{t_{m-1}}(\beta),\ n = 1,\dots,N,\ \beta = 1,\dots,\nu$, are approximated by,

  $$\widehat{Q}_{t_{m-1}}\left(\mathbf{X}_{t_{m-1}}(n)\right) = \mathbb{E}\left[G\left(\mathbf{X}_{t_m}, \alpha^{\beta}_{t_m}\right) \mid \mathbf{X}_{t_{m-1}}(n)\right].$$

  Exploiting the linearity of the expectation operator, using Equation (7), it can be written as:

  $$\widehat{Q}_{t_{m-1}}(\mathbf{X}_{t_{m-1}}(n)) = \sum_{k=1}^{K} \widehat{\alpha}^{\beta}_{t_m}(k)\mathbb{E}\left[\phi_k(\mathbf{X}_{t_m}) \mid \mathbf{X}_{t_{m-1}}(n)\right]. \tag{9}$$

  The vector of basis functions ϕ should ideally be chosen such that the expectations $\mathbb{E}\left[\phi_k(\mathbf{X}_{t_m})|\mathbf{X}_{t_{m-1}}\right]$ are known in closed-form, or have analytic approximations.

  The option value at each exercise time is then given by:

  $$\widehat{V}_{t_{m-1}}\left(\mathbf{X}_{t_{m-1}}(n)\right) = \max\left(h\left(\mathbf{X}_{t_{m-1}}(n)\right), \widehat{Q}_{t_{m-1}}\left(\mathbf{X}_{t_{m-1}}(n)\right)\right).$$

SGBM has been applied to a wide range of problems, high-dimensional Bermudan options with different payoffs in (17), models with jumps in Cong and Oosterlee (2015)(7), for Bermudan swaptions under the Libor market model in Karlsson *et al.* (2016)(19). SGBM has been used for CVA calculations in de Graaf *et al.* (2014)(8) for exposure and potential future exposure calculation, in Qian and Oosterlee (2016)(10) where the underlying follows the Heston Hull-White dynamics, and in Karlsson *et al.* (2016)(18) to compute CVA of Bermudan swaption when the underlying follows Hull-White dynamics. In Qian *et al.* (2017)(9) the suitability of SGBM when different measures are simultaneously used for pricing and risk estimation has been proposed. In Jain *et al.* (2016)(16) the effectiveness of SGBM for hedging KVA, while avoiding nested simulations, has been demonstrated. Leitao and Oosterlee (2015)(21) discuss the approach of massive parallelization using GPUs for SGBM.

4. Sensitivities along paths using SGBM

In this section we describe the approach to compute sensitivities along scenarios with respect to the underlying states and model parameters using SGBM. As the basic *regress-later* Monte Carlo scheme can be seen as a special case of SGBM, where the number of bundles chosen at each time step is equal to one, the approach described here will also be applicable for this scheme.

The method follows a backward iteration, starting from the final expiration time T, where the option value and the sensitivities are trivial to calculate. We emphasize again here that a key difference between SGBM and LSMC is that for SGBM the continuation value is directly computed using the backward iteration as described in Section 2, while LSMC computes it by discounting cashflows as a result of following a (*quasi*)optimal policy. An outcome is that in the case of SGBM, the differentiation of regression coefficients is essential, as it is directly involved in the approximation of the value function, while in the case of LSMC it is not so important as regression is used only to determine the optimal policy.

4.1. *Model parameter-based sensitivities*

The sensitivity of the option value at t_{m-1} with respect to the model parameter θ is given by

$$\begin{aligned}\frac{\partial}{\partial\theta}V_{t_{m-1}}(\mathbf{X}_{t_{m-1}}) &= \left(\frac{\partial}{\partial\theta}h\left(\mathbf{X}_{t_{m-1}}\right)\right)\mathbb{1}_{Q_{t_{m-1}}<h\left(\mathbf{X}_{t_{m-1}}\right)}\\ &+\left(\frac{\partial}{\partial\theta}Q_{t_{m-1}}\left(\mathbf{X}_{t_{m-1}}\right)\right)\mathbb{1}_{Q_{t_{m-1}}\geq h\left(\mathbf{X}_{t_{m-1}}\right)}.\end{aligned}$$

The first term at the right-hand side, the sensitivity of the immediate payoff, is usually trivial to compute. Computing the sensitivity of the continuation value is discussed in detail here. As described in Section 3, the continuation value in SGBM is approximated using Equation (9). The sensitivity of the continuation value for the simulated state $\mathbf{X}_{t_{m-1}}(n)$ in bundle $\mathcal{B}_{t_{m-1}}(\beta)$ is computed as

$$\begin{aligned}\frac{\partial}{\partial\theta}\widehat{Q}_{t_{m-1}}(\mathbf{X}_{t_{m-1}}(n)) &= \frac{\partial}{\partial\theta}\left(\sum_{k=1}^{K}\widehat{\alpha}^{\beta}_{t_m}(k)\mathbb{E}\left[\phi_k(\mathbf{X}_{t_m})\mid\mathbf{X}_{t_{m-1}}(n)\right]\right)\\ &=\sum_{k=1}^{K}\left(\left(\frac{\partial}{\partial\theta}\widehat{\alpha}^{\beta}_{t_m}(k)\right)\mathbb{E}\left[\phi_k(\mathbf{X}_{t_m})\mid\mathbf{X}_{t_{m-1}}(n)\right]\right.\\ &\left.+\ \widehat{\alpha}^{\beta}_{t_m}(k)\frac{\partial}{\partial\theta}\mathbb{E}\left[\phi_k(\mathbf{X}_{t_m})\mid\mathbf{X}_{t_{m-1}}(n)\right]\right) \qquad (10)\end{aligned}$$

In practice, the basis functions are selected such that their conditional moments are either known or can be readily approximated using a numerical scheme. As a result the last term in Equation (10), i.e.

$$\frac{\partial}{\partial \theta}\mathbb{E}\left[\phi_k(\mathbf{X}_{t_m}) \mid \mathbf{X}_{t_{m-1}}(n)\right],$$

can also be readily evaluated.

To arrive at the sensitivities of the regression coefficients, let us first define matrix $\mathbf{A}^{\beta}_{t_m}$ as

$$\mathbf{A}^{\beta}_{t_m} := \begin{bmatrix} \phi_1(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(1))) & \phi_2(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(1))) & \dots & \phi_K(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(1))) \\ \phi_1(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(2))) & \phi_2(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(2))) & \dots & \phi_K(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(2))) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(N_\beta))) & \phi_2(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(N_\beta))) & \dots & \phi_K(\mathbf{X}_{t_m}\mathcal{I}^{\beta}_{t_{m-1}}((N_\beta))) \end{bmatrix}, \tag{11}$$

where $\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(1)), \dots, \mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(N_\beta))$ are the realized states at t_m of the paths in bundle $\mathcal{B}_{t_{m-1}}(\beta)$. Let the corresponding vector of option values at t_m for these paths be defined as

$$\mathbf{V}^{\beta}_{t_m} := \begin{bmatrix} \widehat{V}_{t_m}(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(1))) \\ \widehat{V}_{t_m}(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(2))) \\ \vdots \\ \widehat{V}_{t_m}(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(N_\beta))) \end{bmatrix}. \tag{12}$$

The least squares coefficients for Equation (8) are computed as:

$$\widehat{\alpha}^{\beta}_{t_m} = (\mathbf{A}^{\beta}_{t_m}{}^{\top}\mathbf{A}^{\beta}_{t_m})^{-1}(\mathbf{A}^{\beta}_{t_m}{}^{\top})\mathbf{V}^{\beta}_{t_m}. \tag{13}$$

The derivative of the regression coefficients using the relation from Equation (13) is then given by:

$$\begin{aligned}\frac{\partial \alpha^{\beta}_{t_m}}{\partial \theta} &= \frac{\partial(\mathbf{A}^{\beta}_{t_m}{}^{\top}\mathbf{A}^{\beta}_{t_m})^{-1}}{\partial \theta}(\mathbf{A}^{\beta}_{t_m}{}^{\top})\mathbf{V}^{\beta}_{t_m} \\ &+ (\mathbf{A}^{\beta}_{t_m}{}^{\top}\mathbf{A}^{\beta}_{t_m})^{-1}\frac{\partial \mathbf{A}^{\beta}_{t_m}{}^{\top}}{\partial \theta}\mathbf{V}^{\beta}_{t_m} \\ &+ (\mathbf{A}^{\beta}_{t_m}{}^{\top}\mathbf{A}^{\beta}_{t_m})^{-1}(\mathbf{A}^{\beta}_{t_m}{}^{\top})\frac{\partial \mathbf{V}^{\beta}_{t_m}}{\partial \theta},\end{aligned} \tag{14}$$

where the derivative of the matrix inverse in Equation (14) can be further expanded as

$$\frac{\partial(\mathbf{A}^{\beta}_{t_m}{}^{\top}\mathbf{A}^{\beta}_{t_m})^{-1}}{\partial \theta} = -(\mathbf{A}^{\beta}_{t_m}{}^{\top}\mathbf{A}^{\beta}_{t_m})^{-1}\left(\frac{\partial \mathbf{A}^{\beta}_{t_m}{}^{\top}}{\partial \theta}\mathbf{A}^{\beta}_{t_m} + \mathbf{A}^{\beta}_{t_m}{}^{\top}\frac{\partial \mathbf{A}^{\beta}_{t_m}}{\partial \theta}\right)(\mathbf{A}^{\beta}_{t_m}{}^{\top}\mathbf{A}^{\beta}_{t_m})^{-1}.$$

In order to compute the regression coefficient sensitivities, following the expression in Equation (14), one has to compute, conditional on $\mathcal{F}_{t_{m-1}}$, the quantities $\frac{\partial \mathbf{A}^{\beta}_{t_m}}{\partial\theta}$ and $\frac{\partial \mathbf{V}^{\beta}_{t_m}}{\partial\theta}$.

Here,

$$\frac{\partial \mathbf{A}^{\beta}_{t_m}}{\partial\theta} = \begin{bmatrix} \frac{\partial\phi_1(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(1)))}{\partial \mathbf{X}_{t_m}}\frac{\partial \mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(1))}{\partial\theta} & \cdots & \frac{\partial\phi_K(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}1)))}{\partial \mathbf{X}_{t_m}}\frac{\partial \mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(1))}{\partial\theta} \\ \frac{\partial\phi_1(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(2)))}{\partial \mathbf{X}_{t_m}}\frac{\partial \mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(2))}{\partial\theta} & \cdots & \frac{\partial\phi_K(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(2)))}{\partial \mathbf{X}_{t_m}}\frac{\partial \mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(2))}{\partial\theta} \\ \vdots & \ddots & \vdots \\ \frac{\partial\phi_1(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(N_\beta)))}{\partial \mathbf{X}_{t_m}}\frac{\partial \mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(N_\beta))}{\partial\theta} & \cdots & \frac{\partial\phi_K(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(N_\beta)))}{\partial \mathbf{X}_{t_m}}\frac{\partial \mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(N_\beta))}{\partial\theta} \end{bmatrix},$$

where, conditioned on $\mathcal{F}_{t_{m-1}}$, when the monitoring dates t_m are the same as the discretization steps, then using the discretization scheme in Equation (5) ,

$$\begin{aligned} \frac{\partial \mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(n))}{\partial\theta} &= \frac{\partial}{\partial\theta}F_{m-1}(\mathbf{X}_{t_{m-1}}(\mathcal{I}^{\beta}_{t_{m-1}}(n)), \mathbf{Z}_{t_{m-1}}(\mathcal{I}^{\beta}_{t_{m-1}}(n)), \theta), \\ & n = 1, \dots, N_\beta. \end{aligned} \tag{15}$$

Remark 1 *When the monitoring dates are different from the discretization time steps, for instance, when an exact discretization is not available, the sensitivities in Equation* (15)*, i.e.*

$$\frac{\partial\phi_k(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(n))))}{\partial \mathbf{X}_{t_m}}\frac{\partial \mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(n))}{\partial\theta}, k = 1 \dots, K, n = 1, \dots, N_\beta, \tag{16}$$

can be computed as follows. Let the discretization between t_{m-1} and t_m be denoted by $t_{m-1} = t_{m_0}, \dots, t_{m_l}, \dots, t_{m_L} = t_m$. Denote

$$\begin{aligned} \Theta_{m_{l-1}}(\mathcal{I}^{\beta}_{t_{m_0}}(n)) &= \frac{\partial \mathbf{X}_{t_{m_{l-1}}}(\mathcal{I}^{\beta}_{t_{m_0}}(n))}{\partial\theta}, \\ &= \frac{\partial}{\partial\theta}F_{m_{l-1}}(\mathbf{X}_{t_{m_{l-1}}}(\mathcal{I}^{\beta}_{t_{m_0}}(n)), \mathbf{Z}_{t_{m_{l-1}}}(\mathcal{I}^{\beta}_{t_{m_0}}(n)), \theta), \end{aligned}$$

where $n = 1, \dots, N_\beta$. For ease of notation we drop the path index and write,

$$\begin{aligned} \Theta_{m_{l-1}} &:= \Theta_{m_{l-1}}(\mathcal{I}^{\beta}_{t_{m_0}}(n)) \\ \frac{\partial}{\partial\theta}F_{m_{l-1}}(\mathbf{X}_{t_{m_{l-1}}}(\mathcal{I}^{\beta}_{t_{m_0}}(n)), \mathbf{Z}_{t_{m_{l-1}}}(\mathcal{I}^{\beta}_{t_{m_0}}(n)), \theta) &:= \frac{\partial}{\partial\theta}F_{m_{l-1}}(\mathbf{X}_{t_{m_{l-1}}}, \ , \theta), \end{aligned}$$

then $\frac{\partial \mathbf{X}_{t_{m_L}}}{\partial\theta}$ is computed using the recursion,

$$\Theta_{m_l} = \frac{\partial F_{m_{l-1}}(\mathbf{X}_{t_{m_{l-1}}}, \ , \theta)}{\partial \mathbf{X}_{t_{m_{l-1}}}}\Theta_{m_{l-1}} + \frac{\partial F_{m_{l-1}}(\mathbf{X}_{t_{m_{l-1}}}, \ , \theta)}{\partial\theta}, \tag{17}$$

where $l = 1, \ldots, L$, *with initial condition* $\Theta_{m_0} = 0$. *The recursion in Equation* (17) *can be computed either in the forward or in the adjoint mode, as described in Giles and Glasserman (2005)(12), with the difference that the recursion is restricted between* t_m, *and* t_{m-1} *in* Rolling Adjoints *rather than between* t_M, *and* t_0, *in* Smoking Adjoints*.*

$\mathbf{V}^{\beta}_{t_m}$ is explicitly dependent on the state and model parameters, i.e., $\mathbf{V}^{\beta}_{t_m} := \mathbf{V}^{\beta}_{t_m}(\mathbf{X}_{t_m}, \theta)$. The state $\mathbf{X}_{t_m}|\mathcal{F}_{t_{m-1}}$ itself is again dependent on the model parameters as given by the relation in Equation (5). Therefore,

$$\frac{\partial \mathbf{V}^{\beta}_{t_m}}{\partial \theta}|\mathcal{F}_{t_{m-1}} = \frac{\partial \mathbf{V}^{\beta}_{t_m}}{\partial \mathbf{X}_{t_m}} \frac{\partial \mathbf{X}_{t_m}}{\partial \theta} + \frac{\partial \mathbf{V}^{\beta}_{t_m}}{\partial \theta}, \tag{18}$$

where $\frac{\partial \mathbf{X}_{t_m}}{\partial \theta}|\mathcal{F}_{t_{m-1}}$ is given by Equation (15). When the monitoring dates are not the same as the discretization steps, then

$$\frac{\partial \mathbf{V}^{\beta}_{t_m}}{\partial \mathbf{X}_{t_m}} \frac{\partial \mathbf{X}_{t_m}}{\partial \theta}|\mathcal{F}_{t_{m-1}}$$

can be computed in a forward or an adjoint mode following a methodology analogous to the description in Remark 1. Computing $\frac{\partial \mathbf{V}^{\beta}_{t_m}}{\partial \mathbf{X}_{t_m}}$ is detailed in Section 4.2.

4.2. *Computation of Delta along the paths*

The sensitivity of the option value at t_{m-1} with respect to the $\mathbf{X}_{t_{m-1}}$ is given by

$$\begin{aligned}\frac{\partial V_{t_{m-1}}(\mathbf{X}_{t_{m-1}})}{\partial \mathbf{X}_{t_{m-1}}} &= \left(\frac{\partial h\left(\mathbf{X}_{t_{m-1}}\right)}{\partial \mathbf{X}_{t_{m-1}}}\right) \mathbb{1}_{Q_{t_{m-1}} < h(\mathbf{X}_{t_{m-1}})} \\ &\quad + \left(\frac{\partial Q_{t_{m-1}}\left(\mathbf{X}_{t_{m-1}}\right)}{\partial \mathbf{X}_{t_{m-1}}}\right) \mathbb{1}_{Q_{t_{m-1}} \geq h(\mathbf{X}_{t_{m-1}})}.\end{aligned}$$

The derivative of the immediate payoff h with respect to the state space is usually trivial to compute. Of interest is the computation of the sensitivity of the continuation value function, i.e.,

$$\begin{aligned}\frac{\partial \widehat{Q}_{t_{m-1}}(\mathbf{X}_{t_{m-1}}(n))}{\partial \mathbf{X}_{t_{m-1}}} &= \frac{\partial}{\partial \mathbf{X}_{t_{m-1}}} \left(\sum_{k=1}^{K} \widehat{\alpha}^{\beta}_{t_m}(k) \mathbb{E}\left[\phi_k(\mathbf{X}_{t_m}) \mid \mathbf{X}_{t_{m-1}}(n)\right] \right) \\ &= \sum_{k=1}^{K} \left(\frac{\partial \widehat{\alpha}^{\beta}_{t_m}(k)}{\partial \mathbf{X}_{t_{m-1}}} \mathbb{E}\left[\phi_k(\mathbf{X}_{t_m}) \mid \mathbf{X}_{t_{m-1}}(n)\right] \right. \\ &\quad \left. + \widehat{\alpha}^{\beta}_{t_m}(k) \frac{\partial}{\partial \mathbf{X}_{t_{m-1}}} \mathbb{E}\left[\phi_k(\mathbf{X}_{t_m}) \mid \mathbf{X}_{t_{m-1}}(n)\right] \right),\end{aligned} \tag{19}$$

where, again,

$$\frac{\partial}{\partial \mathbf{X}_{t_{m-1}}} \mathbb{E}\left[\phi_k(\mathbf{X}_{t_m}) \mid \mathbf{X}_{t_{m-1}}(n)\right]$$

is either known in closed-form, or can be readily approximated using a numerical scheme. Following the notation and definitions from Section 4.1, we can write

$$\begin{aligned}
\frac{\partial \alpha^{\beta}_{t_m}}{\partial \mathbf{X}_{t_{m-1}}} &= \frac{\partial ({\mathbf{A}^{\beta}_{t_m}}^{\top} \mathbf{A}^{\beta}_{t_m})^{-1}}{\partial \mathbf{X}_{t_{m-1}}} ({\mathbf{A}^{\beta}_{t_m}}^{\top}) \mathbf{V}^{\beta}_{t_m} \\
&+ ({\mathbf{A}^{\beta}_{t_m}}^{\top} \mathbf{A}^{\beta}_{t_m})^{-1} \frac{\partial {\mathbf{A}^{\beta}_{t_m}}^{\top}}{\partial \mathbf{X}_{t_{m-1}}} \mathbf{V}^{\beta}_{t_m} \\
&+ ({\mathbf{A}^{\beta}_{t_m}}^{\top} \mathbf{A}^{\beta}_{t_m})^{-1} ({\mathbf{A}^{\beta}_{t_m}}^{\top}) \frac{\partial \mathbf{V}^{\beta}_{t_m}}{\partial \mathbf{X}_{t_{m-1}}},
\end{aligned} \tag{20}$$

where the derivative of the matrix inverse in Equation (20) can be further expanded as

$$\frac{\partial ({\mathbf{A}^{\beta}_{t_m}}^{\top} \mathbf{A}^{\beta}_{t_m})^{-1}}{\partial \mathbf{X}_{t_{m-1}}} = -({\mathbf{A}^{\beta}_{t_m}}^{\top} \mathbf{A}^{\beta}_{t_m})^{-1} \left(\frac{\partial {\mathbf{A}^{\beta}_{t_m}}^{\top}}{\partial \mathbf{X}_{t_{m-1}}} \mathbf{A}^{\beta}_{t_m} + {\mathbf{A}^{\beta}_{t_m}}^{\top} \frac{\partial \mathbf{A}^{\beta}_{t_m}}{\partial \mathbf{X}_{t_{m-1}}} \right) ({\mathbf{A}^{\beta}_{t_m}}^{\top} \mathbf{A}^{\beta}_{t_m})^{-1}$$

with

$$\frac{\partial \mathbf{A}^{\beta}_{t_m}}{\partial \mathbf{X}_{t_{m-1}}} = \begin{bmatrix}
\frac{\partial \phi_1(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(1)))}{\partial \mathbf{X}_{t_m}} \frac{\partial \mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(1))}{\partial \mathbf{X}_{t_{m-1}}} & \cdots & \frac{\partial \phi_K(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}} 1)))}{\partial \mathbf{X}_{t_m}} \frac{\partial \mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(1))}{\partial \mathbf{X}_{t_{m-1}}} \\
\frac{\partial \phi_1(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(2)))}{\partial \mathbf{X}_{t_m}} \frac{\partial \mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(2))}{\partial \mathbf{X}_{t_{m-1}}} & \cdots & \frac{\partial \phi_K(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(2)))}{\partial \mathbf{X}_{t_m}} \frac{\partial \mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(2))}{\partial \mathbf{X}_{t_{m-1}}} \\
\vdots & \ddots & \vdots \\
\frac{\partial \phi_1(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(N_\beta)))}{\partial \mathbf{X}_{t_m}} \frac{\partial \mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(N_\beta))}{\partial \mathbf{X}_{t_{m-1}}} & \cdots & \frac{\partial \phi_K(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(N_\beta)))}{\partial \mathbf{X}_{t_m}} \frac{\partial \mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(N_\beta))}{\partial \mathbf{X}_{t_{m-1}}}
\end{bmatrix},$$

where, conditioned on $\mathcal{F}_{t_{m-1}}$ and using the discretization scheme in Equation (5),

$$\begin{aligned}
\frac{\partial \mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(n))}{\partial \mathbf{X}_{t_{m-1}}} &= \frac{\partial F_{m-1}(\mathbf{X}_{t_{m-1}}(\mathcal{I}^{\beta}_{t_{m-1}}(n)), \mathbf{Z}_{t_{m-1}}(\mathcal{I}^{\beta}_{t_{m-1}}(n)), \theta)}{\partial \mathbf{X}_{t_{m-1}}}, \\
& n = 1, \ldots, N_\beta.
\end{aligned} \tag{21}$$

Remark 2 *When the monitoring dates are not the same as the discretization time steps (as a result of a Taylor-based discretization scheme), then the individual sensitivities in Equation* (21)*,* *i.e.*

$$\frac{\partial \phi_k(\mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(n)))}{\partial \mathbf{X}_{t_m}} \frac{\partial \mathbf{X}_{t_m}(\mathcal{I}^{\beta}_{t_{m-1}}(n))}{\partial \mathbf{X}_{t_{m-1}}}, k = 1, \ldots, K, n = 1, \ldots, N_\beta,$$

can be computed as follows. Given the discretization $t_{m-1} = t_{m_0}, \ldots, t_{m_l}, \ldots, t_{m_L} = t_m$*. Denote*

$$\begin{aligned} \Delta_{m_l}(\mathcal{I}^{\beta}_{t_{m_0}}(n)) &= \frac{\partial \mathbf{X}_{t_{m_l}}(\mathcal{I}^{\beta}_{t_{m_0}}(n))}{\partial \mathbf{X}_{t_{m_{l-1}}}}, \\ &= \frac{\partial F_{m_{l-1}}(\mathbf{X}_{t_{m_{l-1}}}(\mathcal{I}^{\beta}_{t_{m_0}}(n)), \mathbf{Z}_{t_{m_{l-1}}}(\mathcal{I}^{\beta}_{t_{m_0}}(n)), \theta)}{\partial \mathbf{X}_{t_{m_{l-1}}}}, \end{aligned}$$

where $n = 1, \ldots, N_\beta$*. For ease of notation we again drop the path index and write*

$$\Delta_{m_l} := \Delta_{m_l}(\mathcal{I}^{\beta}_{t_{m_0}}(n))$$

then $\frac{\partial \phi_k(\mathbf{X}_{t_{m_L}})}{\partial \mathbf{X}_{t_{m_L}}} \frac{\partial \mathbf{X}_{t_{m_L}}}{\partial \mathbf{X}_{t_{m_0}}}$ *is computed using the recursion,*

$$\frac{\partial \phi_k(\mathbf{X}_{t_{m_L}})}{\partial \mathbf{X}_{t_{m_L}}} \Delta_{m_L} \Delta_{m_{L-1}} \ldots \Delta_{m_0}, \tag{22}$$

with the initial condition of Δ_{m_0} *being an identity matrix. The expression in Equation* (22) *can be computed either in the adjoint mode, i.e. from left to right,*

$$\underset{\frac{\partial \phi_k(\mathbf{X}_{t_{m_L}})}{\partial \mathbf{X}_{t_{m_L}}} \Delta_{m_L} \Delta_{m_{L-1}} \ldots \Delta_{m_0}}{\xrightarrow{\hspace{6em}\textit{Adjoint}\hspace{6em}}}$$

or the forward mode, i.e. solving from right to left,

$$\underset{\frac{\partial \phi_k(\mathbf{X}_{t_{m_L}})}{\partial \mathbf{X}_{t_{m_L}}} \Delta_{m_L} \Delta_{m_{L-1}} \ldots \Delta_{m_0}}{\xleftarrow{\hspace{6em}\textit{Forward}\hspace{6em}}}$$

Again the adjoint or forward here needs to be computed at a stretch only between t_{m-1}*, and* t_m*.*

Following the arguments in Section 4.1, $\frac{\partial \mathbf{V}^{\beta}_{t_m}}{\partial \mathbf{X}_{t_{m-1}}}$ is given by

$$\frac{\partial \mathbf{V}^{\beta}_{t_m}}{\partial \mathbf{X}_{t_{m-1}}} = \frac{\partial \mathbf{V}^{\beta}_{t_m}}{\partial \mathbf{X}_{t_m}} \frac{\partial \mathbf{X}_{t_m}}{\partial \mathbf{X}_{t_{m-1}}}. \tag{23}$$

with $\frac{\partial \mathbf{X}_{t_m}}{\partial \mathbf{X}_{t_{m-1}}}$ computed as in Equation (21) when the monitoring dates and discretization steps are the same, or else $\frac{\partial \mathbf{V}^{\beta}_{t_m}}{\partial \mathbf{X}_{t_m}} \frac{\partial \mathbf{X}_{t_m}}{\partial \mathbf{X}_{t_{m-1}}}$ is computed following the description in Remark 2.

4.3. *General algorithm*

Here, we describe the steps of the extended SGBM algorithm for computing the sensitivities along the paths.

Step I: Generating grid points Simulate N independent copies of the sample paths $\{\mathbf{X}_{t_0}, \ldots, \mathbf{X}_{t_M}\}$ of the underlying process using the recursion:

$$\mathbf{X}_{t_m}(n) = F_{m-1}(\mathbf{X}_{t_{m-1}}(n), \mathbf{Z}_{t_{m-1}}(n), \theta),$$

where $n = 1, \ldots, N$ is the index of the path. We also store the corresponding noise for each path, i.e. $\mathbf{Z}_{t_{m-1}}(n)$.

Step II: Option value, parameter and sensitivities at terminal time Compute the option value at terminal time by,

$$V_{t_M}(\mathbf{X}_{t_M}) = \max(h(\mathbf{X}_{t_M}), 0),$$

the sensitivity with respect to the state space, i.e.,

$$\frac{\partial V_{t_M}(\mathbf{X}_{t_M})}{\partial \mathbf{X}_{t_M}} = \frac{\partial h(\mathbf{X}_{t_M})}{\partial \mathbf{X}_{t_M}} \mathbb{1}_{h(\mathbf{X}_{t_M})>0} + 0\mathbb{1}_{h(\mathbf{X}_{t_M})\leq 0},$$

and sensitivity with respect to the model parameters, i.e

$$\frac{\partial V_{t_M}(\mathbf{X}_{t_M})}{\partial \theta} = \frac{\partial h(\mathbf{X}_{t_M})}{\partial \theta} \mathbb{1}_{h(\mathbf{X}_{t_M})>0} + 0\mathbb{1}_{h(\mathbf{X}_{t_M})\leq 0}.$$

Backward recursion The following steps are subsequently performed for each time step, t_m, $m \leq M$, recursively, moving backwards in time, starting from t_M.

Step III: Compute the conditional sensitivities of the state space Compute the sensitivity of $\mathbf{X}_{t_m}(n)$ given $\mathbf{X}_{t_{m-1}}(n)$, with respect to the model parameters using the discretization relation, i.e.

$$\frac{\partial \mathbf{X}_{t_m}}{\partial \theta} = \frac{\partial F_{m-1}(\mathbf{X}_{t_{m-1}}(n), \mathbf{Z}_{t_{m-1}}(n), \theta)}{\partial \theta}.$$

Step IV: Compute the option sensitivity to model parameters Compute the sensitivity of the option value at t_m given $\mathbf{X}_{t_{m-1}}(n)$, with respect to the model parameters, i.e.,

$$
\begin{aligned}
\frac{\partial V_{t_m}(\mathbf{X}_{t_m}(n))}{\partial \theta}|\mathcal{F}_{t_{m-1}} &= \frac{\partial V_{t_m}(\mathbf{X}_{t_m}(n))}{\partial \theta}|\mathcal{F}_{t_m} + \frac{\partial V_{t_m}(\mathbf{X}_{t_m}(n))}{\partial \mathbf{X}_{t_m}}\frac{\partial \mathbf{X}_{t_m}}{\partial \theta} \\
&= \frac{\partial V_{t_m}(\mathbf{X}_{t_m}(n))}{\partial \theta}|\mathcal{F}_{t_m} + \frac{\partial V_{t_m}(\mathbf{X}_{t_m}(n))}{\partial \mathbf{X}_{t_m}}\frac{\partial F_{m-1}(\mathbf{X}_{t_{m-1}}(n), \mathbf{Z}_{t_{m-1}}(n), \theta)}{\partial \theta}.
\end{aligned}
$$

Step V: Bundling The grid points at t_{m-1} are *bundled* into $\mathcal{B}_{t_{m-1}}(1), \dots, \mathcal{B}_{t_{m-1}}(\nu)$ non-overlapping partitions. This step is similar to the one described in Section 3. For each bundle $\mathcal{B}_{t_{m-1}}(\beta),\ \beta = 1, \dots, \nu$, the following steps are performed.

- **Step V(a): *Regress-later* for each bundle**

  This step is as described in Section 3. Corresponding to each bundle $\mathcal{B}_{t_{m-1}}(\beta),\ \beta = 1, \dots, \nu$, a parameterized value function is obtained using least squares regression, i.e.,

  $$V_{t_m}(\mathbf{X}_{t_m}) \approx G(\mathbf{X}_{t_m}, \alpha^{\beta}_{t_m}) = \sum_{k=1}^{K} \alpha^{\beta}_{t_m}(k)\phi_k(\mathbf{X}_{t_m}),$$

  The coefficients from the least squares regression are determined by solving:

  $$\widehat{\alpha}^{\beta}_{t_m} = (\mathbf{A}^{\beta\ \top}_{t_m}\mathbf{A}^{\beta}_{t_m})^{-1}(\mathbf{A}^{\beta\ \top}_{t_m})\mathbf{V}^{\beta}_{t_m},$$

  where $\mathbf{A}^{\beta}_{t_m}$ and $\mathbf{V}^{\beta}_{t_m}$ are defined as Equations (11) and (12), respectively.

- **Step V(b): Compute conditional sensitivities for matrix $\mathbf{A}^{\beta}_{t_m}$**

  Compute the sensitivity of $\mathbf{A}^{\beta}_{t_m}$ conditioned on the information at t_{m-1}, with respect to the state space at t_{m-1} as in Equation (21) and with respect to the model parameters as in Equation (15).

- **Step V(c): Compute regression coefficient sensitivities**

  Compute the sensitivity of the regression coefficients, $\widehat{\alpha}^{\beta}_{t_m}$, with respect to the state space using Equation (20) and with respect to the model parameters as given by Equation (14).

- **Step V(d): Compute the continuation and option values at t_{m-1}**

  As described in Section 3, the continuation values for $\mathbf{X}_{t_{m-1}}(n) \in \mathcal{B}_{t_{m-1}}(\beta),\ n = 1, \dots, N,\ \beta = 1, \dots, \nu$, are approximated using Equation (9).

  The vector of basis functions ϕ should ideally be chosen such that the expectation $\mathbb{E}\left[\phi_k(\mathbf{X}_{t_m})|\mathbf{X}_{t_{m-1}}\right]$ is known in closed-form, or has an analytic approximation.

  The option value is then given by:

  $$\widehat{V}_{t_{m-1}}\left(\mathbf{X}_{t_{m-1}}(n)\right) = \max\Big(h\left(\mathbf{X}_{t_{m-1}}(n)\right), \widehat{Q}_{t_{m-1}}\left(\mathbf{X}_{t_{m-1}}(n)\right)\Big).$$

- **Step V(e): Compute the sensitivity of the continuation values at t_{m-1}**

The sensitivity of the continuation value with respect to the model parameters for the paths at t_{m-1} is computed using the relation in Equation (10) and with respect to the state space is computed using the relation in Equation (19), where

$$\frac{\partial}{\partial\theta}\mathbb{E}\left[\phi_k(\mathbf{X}_{t_m}) \mid \mathbf{X}_{t_{m-1}}(n)\right],$$

and

$$\frac{\partial}{\partial\mathbf{X}_{t_{m-1}}}\mathbb{E}\left[\phi_k(\mathbf{X}_{t_m}) \mid \mathbf{X}_{t_{m-1}}(n)\right]$$

are approximated analytically or by means of a numerical scheme.

- **Step V(f): Compute the sensitivity of the option values at t_{m-1}**

  The sensitivity of the option values at t_{m-1} for the n-*th* path with respect to the model parameters, i.e., $\frac{\partial V_{t_{m-1}}(\mathbf{X}_{t_{m-1}})}{\partial\theta}|\mathcal{F}_{t_{m-1}}$, is then computed as:

$$\begin{aligned}\frac{\partial}{\partial\theta}V_{t_{m-1}}(\mathbf{X}_{t_{m-1}}(n)) &= \left(\frac{\partial}{\partial\theta}h\left(\mathbf{X}_{t_{m-1}(n)}\right)\right)\mathbb{1}_{Q_{t_{m-1}}(\mathbf{X}_{t_{m-1}}(n))<h\left(\mathbf{X}_{t_{m-1}}(n)\right)}\\ &+\left(\frac{\partial}{\partial\theta}Q_{t_{m-1}}\left(\mathbf{X}_{t_{m-1}}(n)\right)\right)\mathbb{1}_{Q_{t_{m-1}}(\mathbf{X}_{t_{m-1}}(n))\geq h\left(\mathbf{X}_{t_{m-1}}(n)\right)},\end{aligned}$$

  while, similarly, the derivative with respect to the state space is computed as

$$\begin{aligned}\frac{\partial V_{t_{m-1}}(\mathbf{X}_{t_{m-1}}(n))}{\partial\mathbf{X}_{t_{m-1}}} &= \left(\frac{\partial h\left(\mathbf{X}_{t_{m-1}(n)}\right)}{\partial\mathbf{X}_{t_{m-1}}}\right)\mathbb{1}_{Q_{t_{m-1}}(\mathbf{X}_{t_{m-1}}(n))<h\left(\mathbf{X}_{t_{m-1}}(n)\right)}\\ &+\left(\frac{\partial Q_{t_{m-1}}\left(\mathbf{X}_{t_{m-1}}(n)\right)}{\partial\mathbf{X}_{t_{m-1}}}\right)\mathbb{1}_{Q_{t_{m-1}}(\mathbf{X}_{t_{m-1}}(n))\geq h\left(\mathbf{X}_{t_{m-1}}\right)}.\end{aligned}$$

The recursion continues until we reach t_0.

Remark 3 *A lower bound for the option value can be obtained by means of the so-called SGBM path estimator formulation, as described in (17). The path estimator value is the expected discounted cashflow resulting from a (*quasi-*)optimal policy. In the SGBM setting, the policy is in the form of the regression functions G, as in Equation (7). For an out-of-sample set of paths*[1]*, the continuation value is implicitly obtained using the G functions. The continuation value, in turn, is used to determine the first early-exercise time step for each out-of-sample path. The expectation of resulting discounted cashflows would then be a lower bound for the option price. The path estimator based on the t_0-Greeks is the expected pathwise sensitivity of the discounted cashflows resulting from the (quasi)optimal policy. A brute force computation would be to apply the* bump-and-revalue *approach to the path estimator, where a fixed (quasi)optimal policy from un-bumped scenarios would be used to determine the early-exercise cashflows for the bumped scenarios. Unlike the option values, the t_0 sensitivities obtained using the path estimator are not necessarily lower or upper bounds of the true sensitivities, though.*

[1] not the same set of paths used to obtain the G functions

5. Numerical examples

In order to illustrate the performance of the method we start with a very basic example of a European put option on a stock index that follows a Geometric Brownian Motion (GBM). As the option values and their sensitivities for this case are known in closed-form, we can numerically study the error along the paths for SGBM. We then consider the corresponding Bermudan option case, for which reference sensitivities at t_0 are also computed using the COS method. We then consider high-dimensional payoffs, in particular, a Bermudan spread option on two assets for which reference values are computed using a *BR* approach.

5.1. *European option on a single asset*

To validate the method, we look at a case where the option sensitivities are known in closed-form to compare against SGBM generated sensitivities along a path. The stochastic differential equation (SDE) for the risk neutral asset price dynamics following the GBM process is given by:

$$d\mathbf{X}_t = r\mathbf{X}_t dt + \sigma \mathbf{X}_t dW_t \tag{24}$$

where r is the risk free rate of return, σ is the annualized volatility (both chosen to be constant) and W_t is the standard Brownian motion. In order to simulate the above process, one can either use an Euler discretization or an exact simulation scheme. For our numerical analysis we use the exact discretization scheme, i.e.,

$$\log(X_{t_m}(n)) = \log(X_{t_{m-1}}(n)) + \left(r - \frac{\sigma^2}{2}\right)\Delta t + \sigma\sqrt{\Delta t} Z_{t_{m-1}}(n), \tag{25}$$

where $n = 1, \ldots N$, gives the path index, $t_0, \ldots, t_m, \ldots, t_M = T$, is the set of equally spaced monitoring dates with T being final expiration time, and $\Delta t = t_m - t_{m-1}$. For regression we use polynomials of the logarithm of the asset values up to order three, i.e.

$$\phi_k\left(\mathbf{X}_{t_m}(n)\right) = \left(\log\left(\mathbf{X}_{t_m}(n)\right)\right)^{k-1}, k = 1, \ldots, K = 4. \tag{26}$$

Note that the conditional expectations of the basis functions are known as the conditional distribution $\log(\mathbf{X}_{t_m})|\mathbf{X}_{t_{m-1}}(n)$ is normal with mean $\log(\mathbf{X}_{t_{m-1}}(n)) + (r - \frac{\sigma^2}{2})\Delta t$, and standard deviation $\sigma\sqrt{(\Delta t)}$.

For the European option case, as intermediate exercise is not allowed, the approximated option values at the monitoring dates is equal to the approximated continuation value, i.e.

$$\widehat{V}_{t_{m-1}}\left(\mathbf{X}_{t_{m-1}}(n)\right) = \widehat{Q}_{t_{m-1}}\left(\mathbf{X}_{t_{m-1}}(n)\right),$$

as opposed to maximum of continuation value and the intrinsic value.

5.1.1. **Delta along the paths.** For the set of simulated scenarios, we compute the Delta along the scenarios of a European put option using the SGBM, as described in Section 4.3, and compare

Set I	$\mathbf{X}_{t_0}$	36, 40, 44
	σ	10%, 20%,40%
	r	0.06
	Strike K	40
	M	50
	T	1 year
Set II	$\mathbf{X}_{t_0} := \{S^1_{t_0}, S^2_{t_0}\}$	[100, 100]
	$\sigma := \{\sigma^1, \sigma^2\}$	[15% 15%]
	r	0.03
	Strike K	5
	M	8
	ρ_{12}	0.5
	T	1 year

Table 1. Parameters used for the different experiments in Section 5.

against the closed-form Black-Scholes (BS) Deltas. We consider different moneyness levels, and study the convergence for varying numbers of bundles and basis functions. The parameter set used for this case is provided in Table 5.1.1. For all examples we use 90,000 scenarios.

In Figure 1 we compare Deltas along the paths computed using SGBM for different time slices against their corresponding closed-form BS Delta values. The SGBM Deltas have a highly satisfactory match with most of the BS Delta values, as can be inferred from the “near one slope” of the linear fit, the reasonably low value of the y-intercept and the small residual errors. The non-smooth nature of the option value function at expiry complicates finding a good fit using a linear regression, the residual errors are (relatively) high towards the expiry.

We next consider the convergence of the option Deltas in relation to the number of bundles. The bundling scheme used here is recursive bifurcation (see (17) for details), with the number of bundles varying from $2^0, \ldots, 2^4$. We look at the mean absolute error (MAE) for different monitoring dates, with increasing numbers of bundles for the same set of paths. The MAE at time step t_m is calculated as:

$$\text{MAE}_{t_m} = \frac{1}{N}\sum_{n=1}^{N} \text{abs}\left(\frac{\partial \widehat{V}_{t_m}(\mathbf{X}_{t_m}(n))}{\partial \mathbf{X}_{t_m}} - \frac{\partial V_{t_m}(\mathbf{X}_{t_m}(n))}{\partial \mathbf{X}_{t_m}}\right),$$

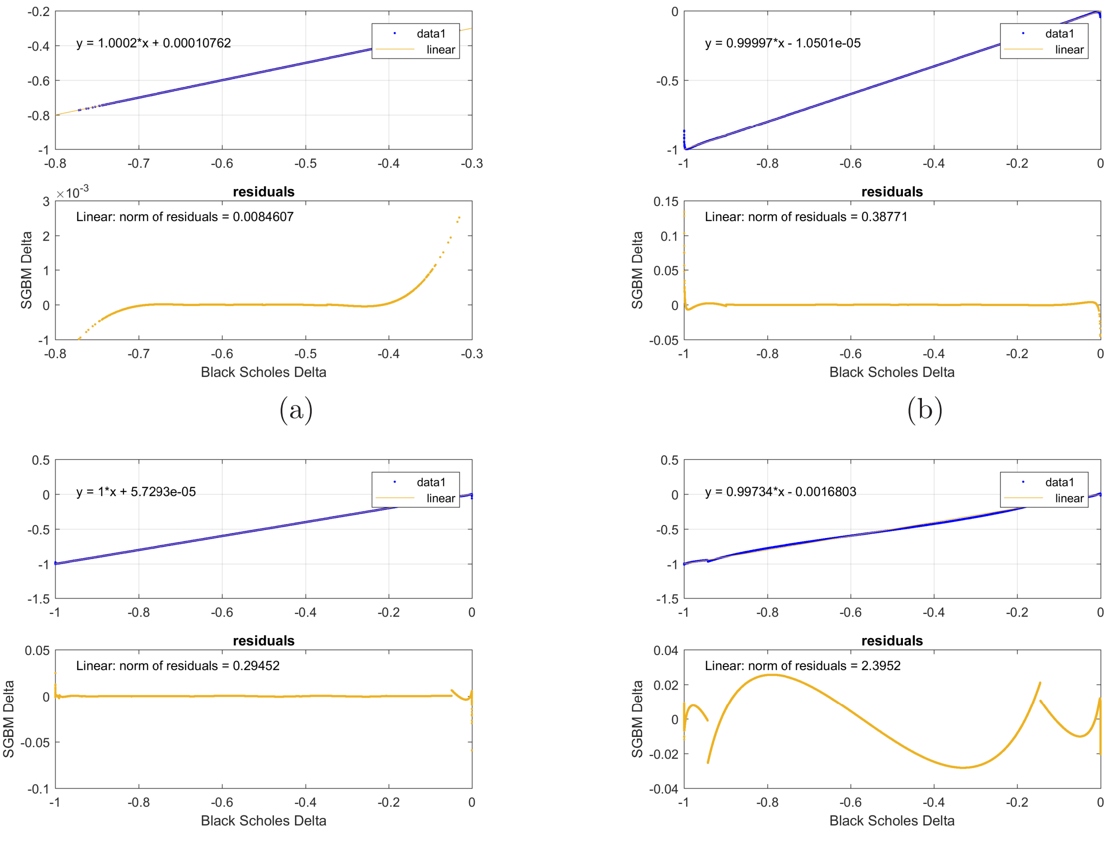
where $\frac{\partial V_{t_m}(\mathbf{X}_{t_m}(n))}{\partial \mathbf{X}_{t_m}}$ is the BS Delta calculated in closed-form.

Figure 2 shows the convergence of the errors for different monitoring dates when the given set of paths is partitioned by an increasing number of bundles in SGBM.

In Figure 3 the effect of increasing the number of basis functions, when all other factors are kept constant is demonstrated.

As the method converges with increasing numbers of bundles and with the basis functions, we argue that SGBM has an advantage over a straightforward *regress-later* scheme which is equivalent to SGBM however with just one bundle, and plain partitioning schemes that are equivalent to SGBM with only the constant as the basis function for regression.

In Table 2 we report the t_0 Delta values for different moneyness levels and corresponding BS Deltas.



(a)

(b)

(c)

(d)

Figure 1. SGBM Deltas plotted against BS Deltas with the corresponding residuals from a linear fit at (a) $t_1 = 0.02$ years (b) $t_{20} = 0.4$ years, (c) $t_{35} = 0.7$ years, (d) $t_{49} = 0.98$ years. Parameters from Table 5.1.1 are used. The basis functions used are defined in Equation (26) with $K = 4$, and the number of bundles at each time step is 2^3. The t_0 BS Delta is -0.55045 and computed using SGBM is -0.55046.

$\mathbf{X}_{t+0}$	SGBM Delta (s.e.)	BS Delta	error	SGBM Vega (s.e.)	BS Vega	error	SGBM Gamma (s.e.)	BS Gamma	error
36	-0.5504 (0.2e-5)	-0.5504	0.1e-4	14.2526 (0.0005)	14.2469	0.0057	0.0550 (0.2e-5)	0.0550	-0.03e-4
40	-0.3445 (0.1e-5)	-0.3445	0.1e-4	14.7399 (0.0006)	14.7308	0.0091	0.0460 (0.2e-5)	0.0460	-0.06e-4
44	-0.1903 (0.4e-5)	-0.1903	0.4e-4	11.9702 (0.0007)	11.9542	0.0160	0.0309 (0.2e-5)	0.0309	-0.12e-4

Table 2. The t_0 Delta, Vega, and Gamma values computed using SGBM. The values in brackets are corresponding standard errors for SGBM for 30 trials. The basis functions for SGBM are as defined in Equation (26) and the number of bundles used are 2^3 at each time step.

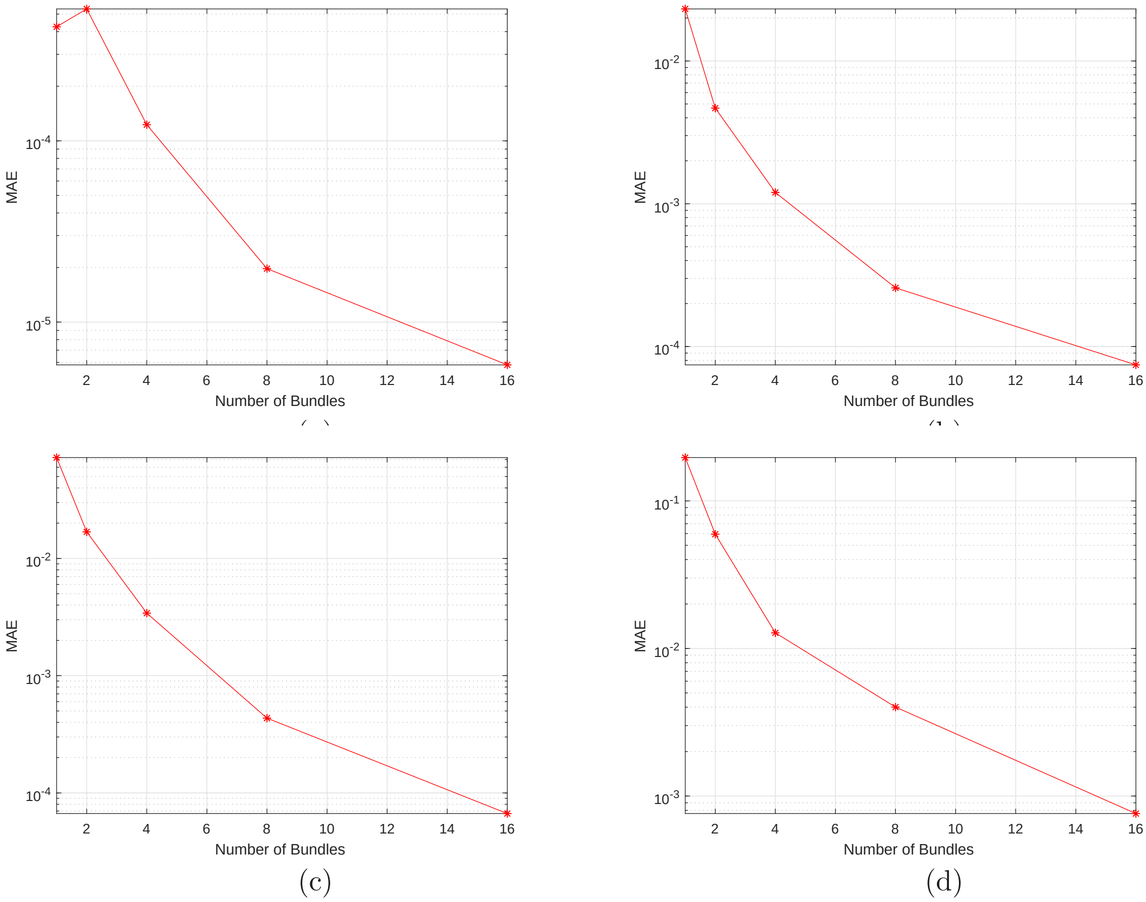
5.1.2. Vega along the paths.

Here we study the quality of the Vegas along the paths computed using SGBM against the closed-form BS Vegas. Figure 4 compares the SGBM Vegas against the corresponding BS Vegas for different monitoring dates. We see that reasonably accurate approximations of these Vegas can be obtained, with an improved accuracy as we approach t_0.

We study the convergence of the SGBM Vega for this case by looking at the MAE at different monitoring dates, for a given set of paths, when there is an increasing number of bundles, in Figure 5, and an increasing number of basis functions, in Figure 6, respectively.

In Figure 3 the effect of increasing the number of basis functions, when all the other factors are kept constant is demonstrated.

Finally, we look at the t_0 SGBM Vega values for different moneyness levels and compare them against the BS values in Table 2. Clearly, the SGBM Vegas are reasonably accurate for the different



(c)

(d)

Figure 2. The MAE of SGBM for an increasing number of bundles at different monitoring dates (a) $t_1 = 0.02$ years (b) $t_{20} = 0.4$ years, (c) $t_{35} = 0.7$ years, (d) $t_{49} = 0.98$ years. The basis functions used are given by Equation (26).

moneyness levels.

The combined computational time for the Vega and Deltas along the 90,000 scenarios for 50 monitoring dates was less than 5 seconds on an Intel Quad Core processor with a 4 GB RAM.

With the help of this basic one-dimensional example we were able to give a detailed path analysis of the Greeks using SGBM. For the remaining problems we look at the t_0 Greeks for SGBM, the reference values of which are computed using a *BR* approach.

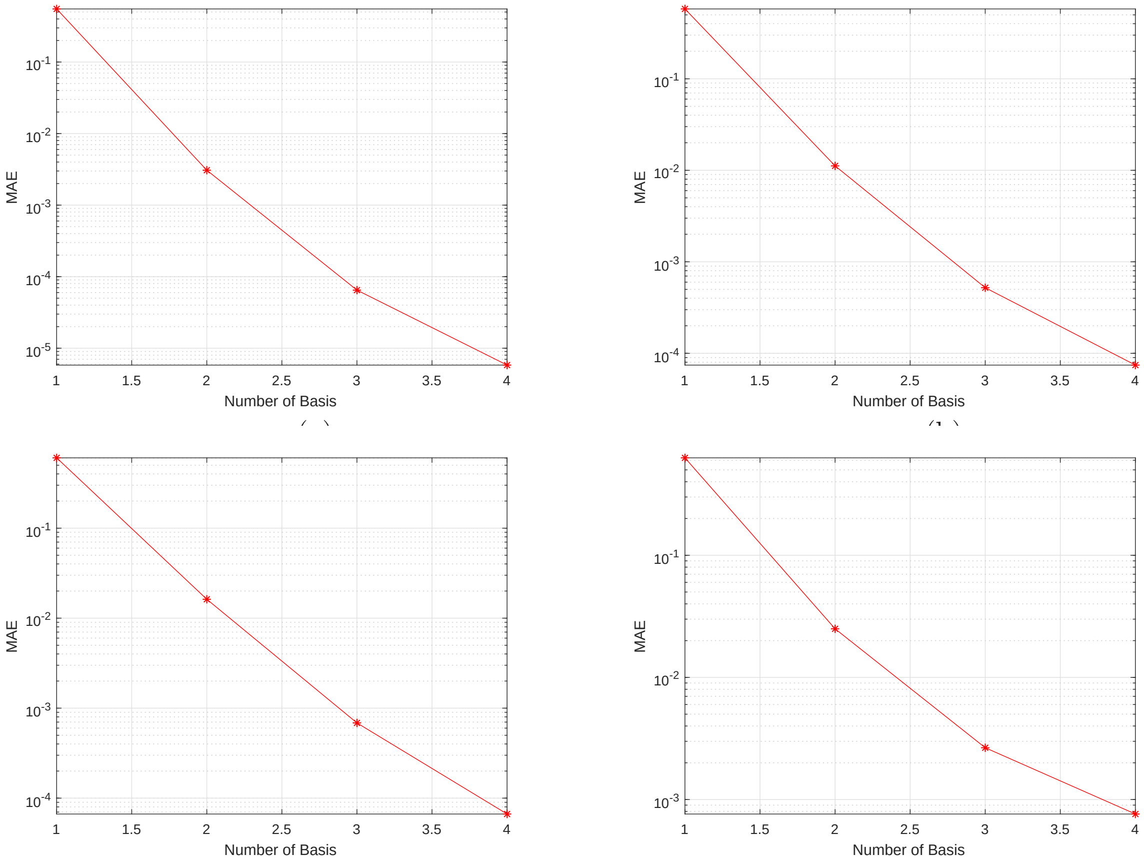
5.1.3. **Gamma along the paths.**

The second order sensitivity with respect to the underlying state can be computed using SGBM, in a manner similar to the Deltas along the paths. It would require additionally to compute the second order derivatives of the conditional expectations of the basis, i.e.

$$\frac{\partial^2 \mathbb{E}\left[\phi_k(\mathbf{X}_{t_m})\big|(\mathbf{X}_{t_{m-1}}(n))\right]}{\partial \mathbf{X}^2_{t_{m-1}}}$$

again something that can be done analytically or numerically.

We again look at the Gamma values along the path for different time slices. Figure 7 compares the SGBM Gamma values against the corresponding closed-form BS Gammas for different monitoring



(c) (d)

Figure 3. The MAE of SGBM for an increasing number of basis functions at different monitoring dates (a) $t_1 = 0.02$ years (b) $t_{20} = 0.4$ years, (c) $t_{35} = 0.7$ years, (d) $t_{49} = 0.98$ years. The basis functions used are defined by Equation (26), where the number of basis functions are modified by varying the value of K from 1 to 5.

dates. We see that reasonably accurate approximations of these Gammas can be obtained, with an improved accuracy as we approach t_0.

SGBM is able to compute with high accuracy the t_0 Gamma values, as can be seen from the results in Table 2.

5.1.4. **Impact of inexact discretization scheme.** We briefly study through numerical experiments the convergence of the method when an Euler discretization scheme is used for the simulation of the scenarios instead of an exact scheme as given in Equation (25). An Euler discretization of the SDE given by Equation (24) is

$$\mathbf{X}_{t_{m_l}}(n) = \mathbf{X}_{t_{m_{l-1}}}(n) + r\mathbf{X}_{t_{m_{l-1}}}(n)\Delta t + \sigma\sqrt{\Delta t}\mathbf{X}_{t_{m_{l-1}}} Z_{t_{m_{l-1}}}(n), \tag{27}$$

where $n = 1, \ldots, N$, gives the path index, $t_0, \ldots, t_m, \ldots, t_M = T$ are the monitoring dates, and the period between each monitoring dates t_{m-1}, and t_m is discretized as $t_{m-1} = t_{m_0}, \ldots, t_{m_l}, \ldots, t_{m_L} = t_m$. For simplicity we take equally spaced time discretization with $\Delta t = t_{m_l} - t_{m_{l-1}}$.

Table 3 gives results corresponding to an increasing number of steps used for the Euler discretization, for the option value, Delta , Vega and Gamma at t_0, using SGBM extended.

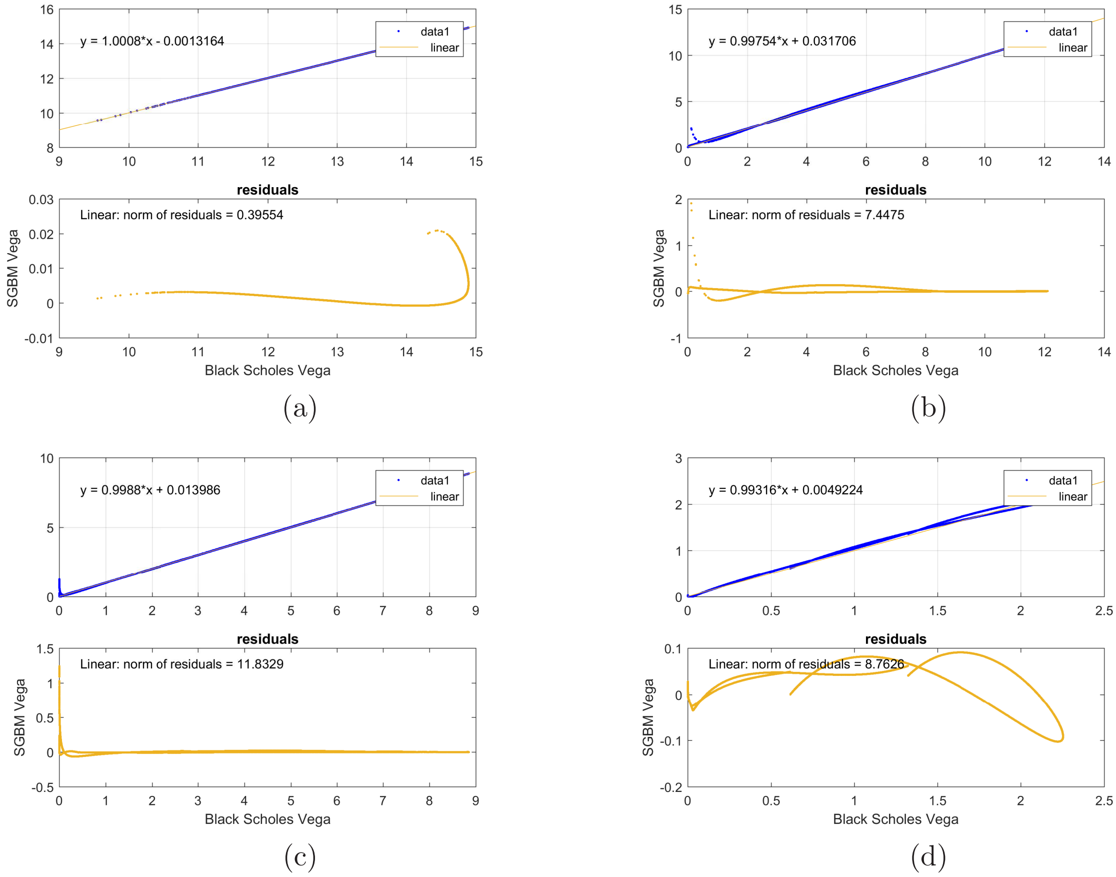


Figure 4. SGBM Vegas plotted against BS Vegas with the corresponding residuals from a linear fit at (a) $t_1 = 0.02$ years (b) $t_{20} = 0.4$ years, (c) $t_{35} = 0.7$ years, (d) $t_{49} = 0.98$ years. Parameters from Table 5.1.1 are used. The t_0 BS Vega is 14.2469 and computed using SGBM is 14.2475

Euler Steps	Option Value (s.e.)	Delta (s.e.)	Vega (s.e.)	Gamma (s.e.)
5	3.8484 (0.0015)	-0.5483 (0.0003)	14.2638 (0.0228)	0.0553 (0.0001)
10	3.8459 (0.0010)	-0.5495 (0.0001)	14.2611 (0.0175)	0.0551 (0.0001)
20	3.8449 (0.0011)	-0.5500 (0.0002)	14.2649 (0.0196)	0.0550 (0.0001)
40	3.8443 (0.0011)	-0.5502 (0.0002)	14.2527 (0.0218)	0.0550 (0.0002)
Exact (∞)	3.8444 (0.0003)	-0.5505 (0.0000)	14.2493 (0.0037)	0.0549 (0.0000)
BS	3.8443	-0.5505	14.2469	0.0550

Table 3. The t_0 price and Greeks for a European option when an inexact Euler discretization scheme is used for simulating the Monte Carlo scenarios and SGBM extended is used for pricing and for computing Greeks along the paths. The corresponding values from an exact discretization and closed for BS is also presented. The initial asset value is 36, and other parameters are taken from Set I in Table 5.1.1.

5.2. *Bermudan option on a single asset*

We next consider a Bermudan put option on a single asset with 50 equally spaced exercise opportunities. The exercise dates coincide with the monitoring dates specified in Section 5.1. The parameters used for this case are also specified in Table 5.1.1. The reference Vega for this case can be computed by the COS method (Ruijter and Oosterlee (2012)(26)).

For all the experiments here we simulate 90,000 scenarios, using an exact discretization scheme

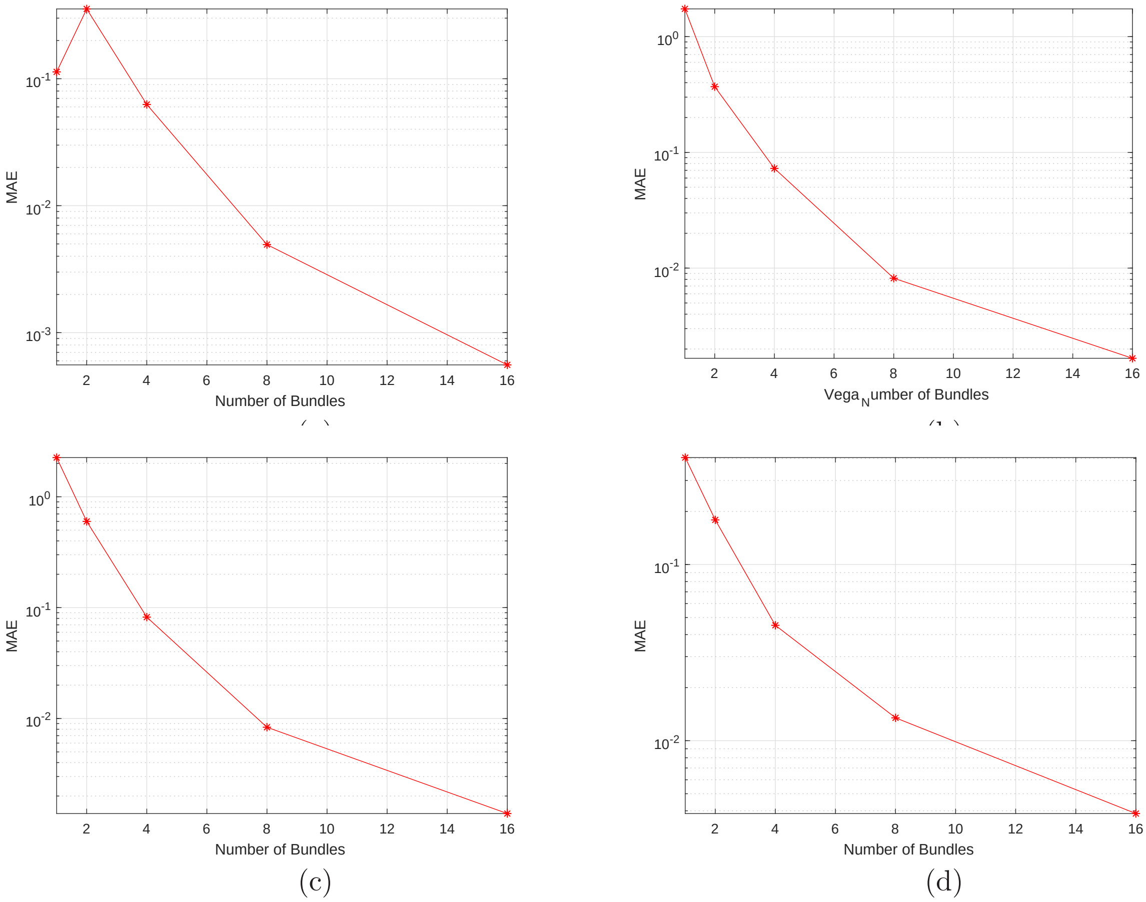


Figure 5. The MAE of SGBM Vega for an increasing number of bundles at different monitoring dates (a) $t_1 = 0.02$ years (b) $t_{20} = 0.4$ years, (c) $t_{35} = 0.7$ years, (d) $t_{49} = 0.98$ years. The basis functions used were $\phi_k(\mathbf{X}_{t_m}) = (log(\mathbf{X}_{t_m}))^{k-1}$, $k = 1, \dots 4$.

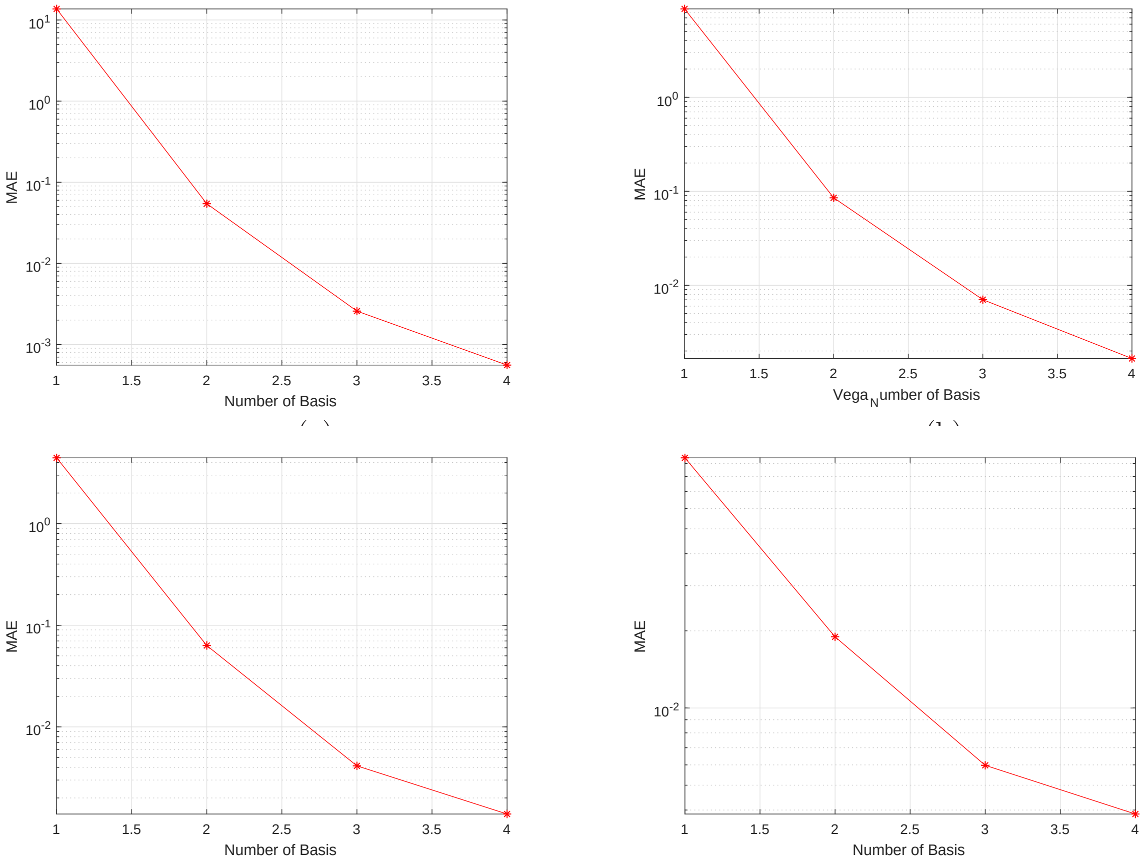
$\mathbf{X}_{t_0}$	COS Delta	SGBM Delta (s.e.)	Error SGBM	LSMC1 Delta (s.e.)	Error LSMC1	LSMC2 Delta (s.e.)	Error LSMC2
36	-0.6959	-0.6957 (0.6e-5)	-0.0001	-0.7118 (0.0213)	0.0159	-0.9729 (0.227)	-0.2770
40	-0.4040	-0.4044 (0.5e-5)	0.0003	-0.4021 (0.0190)	-0.0019	-0.4631 (0.033)	-0.0591
44	-0.2136	-0.2145 (0.9e-5)	0.0009	-0.2277 (0.0080)	0.0141	-0.2532 (0.031)	-0.0396

Table 4. t_0 Delta values for Bermudan put option on a single asset for different initial asset prices. The values in brackets are the standard errors from thirty trials. The reference value is computed by the COS method.

as given by Equation (25). We use a recursive bifurcation scheme for bundling with 2^3 bundles at each time step. Similar to the European option case, as basis functions we use the polynomials up to order three of the log-asset price, as given by Equation (26).

Table 4 reports the SGBM computed Delta for different moneyness levels.

For a comparison we look at the LSMC *BR* prices. We here use two approaches to compute the *BR* sensitivities. The first approach is where the policy (the regression coefficients) is computed for the non-bumped scenario, and this policy is applied to the bumped up and down scenarios for optimal early-exercise. We call this approach “LSMC1”. The second approach is where a separate policy (or regression coefficients) is computed for the bump up and bump down scenarios. While computing the option price for bumped up we use the corresponding policy to determine the optimal early-exercise and cashflows thereof. Similarly, the bumped down option price is based on the policy corresponding to the bumped down scenarios. We represent this approach by “LSMC2” in the results.



(c) (d)

Figure 6. The MAE of SGBM for an increasing number of basis functions at different monitoring dates (a) $t_1 = 0.02$ years (b) $t_{20} = 0.4$ years, (c) $t_{35} = 0.7$ years, (d) $t_{49} = 0.98$ years. The number of bundles were fixed to 2^3 at each time step.

$\mathbf{X}_{t_0}$	COS	SGBM	Error	LSMC1	Error	LSMC2	Error
	Vega	Vega (s.e.)	SGBM	Vega (s.e.)	LSMC1	Vega (s.e.)	LSMC2
36	10.9550	10.9202 (0.001)	-0.0348	11.0995 (0.070)	0.1445	10.7341 (0.231)	-0.2209
40	14.7471	14.7520 (0.001)	0.0049	14.8909 (0.099)	0.1438	14.7301 (0.057)	-0.0170
44	12.5243	12.6167 (0.003)	0.0924	12.5561 (0.062)	0.0318	12.5369 (0.051)	0.0126

Table 5. t_0 Vega values for Bermudan put option on a single asset for different initial asset prices. The values in brackets are the standard errors from thirty trials. The reference values are computed by the COS method.

It is clear that relative to these approaches, the SGBM Deltas are more accurate with significantly lower variance in the corresponding results. LSMC1 comes out as a better choice than LSMC2, especially for in-the-money options. The intuition behind the better performance of LSMC1 in this case is that the optimal policy ideally should not be affected by the choice of the initial asset price, which is enforced by design in LSMC1. In the case of LSMC2 we inadvertently capture errors in the approximated optimal policy while computing the Delta.

In Table 5 we compare the t_0 Vega calculated using SGBM against the LSMC *BR* Vega. The reference value is obtained by the COS method with an accurate setting. Again the bias and standard errors for the SGBM Vegas are much lower than those obtained from either the LSMC1 or LSMC2 *BR* approaches.

For this case both LSMC1 and LSMC2 have similar accuracy, although LSMC1 seems better for the in-the-money case, while LSMC2 is marginally better for the at-the-money and out-of-the-money cases.

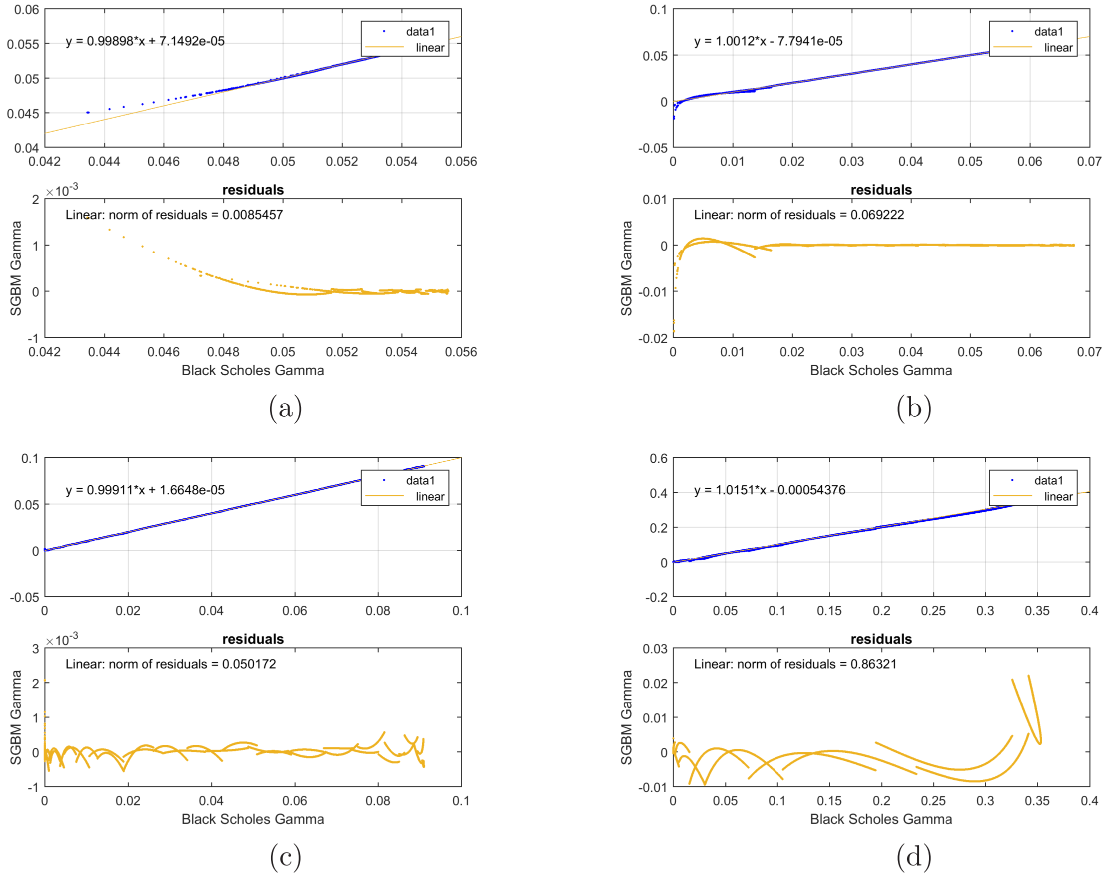


Figure 7. SGBM Gammas plotted against BS Gammas with the corresponding residuals from a linear fit at (a) $t_1 = 0.02$ years (b) $t_{20} = 0.4$ years, (c) $t_{35} = 0.7$ years, (d) $t_{49} = 0.98$ years. Parameters from Table 5.1.1 are used. The basis functions used are defined by Equation (26) and the number of bundles at each time step are 2^5. The t_0 BS Gamma is 0.0550 and computed using SGBM is 0.0550.

σ	COS Vega	SGBM Vega (s.e.)	Error SGBM	LSMC1 Vega (s.e.)	Error LSMC1	LSMC2 Vega (s.e.)	Error LSMC2
10%	13.3607	13.4023 (0.002)	0.0416	13.5261 (0.062)	0.1652	13.2853 (0.066)	-0.0754
20%	14.7471	14.7505 (0.001)	0.0034	14.9312 (0.084)	0.1841	14.7301 (0.057)	-0.0170
40%	15.0553	15.0534 (0.002)	-0.0019	15.1889 (0.104)	0.1336	15.1151 (0.087)	0.0598

Table 6. t_0 Vega values for Bermudan put option on a single asset for different asset volatilities. The values in brackets are the standard errors from thirty trials. The initial asset value is $\mathbf{X}_{t_0} = 40$. The reference values are computed by the COS method.

In Table 6 we report the performance of the SGBM Vega against LSMC1 and LSMC2 *BR* Vega for different initial volatilities. The SGBM results are more stable for increasing asset volatilities.

For this case the LSMC2 seems a better choice than LSMC1. LSMC in general computes a quasi-optimal policy, which is implicitly dependent on the asset volatility. Part of the option Vega values is attributed to the change in cashflows due to a Delta change in the asset volatility with the quasi-optimal policy held constant, while the other part is attributed to changes in the cashflows due to the change in the quasi-optimal policy as a result of Delta change in the asset process volatility. In this regard the approach proposed by both (2) and (6) of differentiating the LSMC regression coefficients may be an efficient way for computing parameter related sensitivities using LSMC.

$\mathbf{X}_{t_0}$	COS Vega	SGBM Vega (s.e.)	Error SGBM	LSMC1 Vega (s.e.)	Error LSMC1	LSMC2 Vega (s.e.)	Error LSMC2
34.5	6.7944	6.7572 (0.0008)	-0.0372	7.0621 (0.212)	0.2677	6.8663 (0.433)	0.0719
35	8.3838	8.3425 (0.001)	0.0414	8.6211 (0.119)	0.2374	8.0762 (0.149)	-0.3075
35.5	9.7714	9.7317 (0.001)	0.0397	10.2243 (0.103)	0.4529	9.4508 (0.161)	-0.3206

Table 7. t_0 Vega values for Bermudan put option on a single asset for a case where the initial asset price is close to the early-exercise boundary. The values in brackets are the standard errors from thirty trials. The reference values are computed by the COS method.

5.2.1. **Unstable LSMC BR Vegas.** We present a test case for which the Vegas computed using LSMC were not stable. In particular, when the initial asset price is close to the early-exercise boundary, the *BR* Vegas using LSMC2 have significant noise. On the other hand, although LSMC1 Vegas have relatively low standard errors, the bias was high in comparison to LSMC2. Therefore one should apply the bump and revalue scheme for Vega calculation using LSMC with due precaution when the initial state space is close to the computed early-exercise boundary. The SGBM path Vegas didn't face any stability issue for this case.

Table 7 reports the Vegas as the initial asset price gets closer to the early-exercise boundary. The standard error for Vega, computed from 30 trials with different initial seeds, by LSMC2 “jumps” when the initial asset price is closest to the early-exercise boundary, i.e $\mathbf{X}_{t_0} = 34.5$, in our example. The minimum and maximum values of Vega from 30 trials for LSMC2 were 4.64 and 9.96, respectively. Although for LSMC1 we don't see this in the standard error, the mean of the trials for LSMC1 seems significantly biased. For the 30 trials the minimum and maximum Vega were 6.35 and 8.25, respectively. For SGBM the results are not affected by the proximity to the early-exercise boundary. The minimum and maximum Vegas from 30 trials for SGBM were 6.74 and 6.76. The reference option Vegas are computed using the COS method.

5.3. *Bermudan spread option on two assets*

We next consider a two-dimensional spread option (based on two assets). The asset prices are assumed to follow correlated geometric Brownian motion processes, i.e.

$$\frac{d\mathbf{X}_t}{\mathbf{X}_t} = r dt + \sigma d\mathbf{W}_t, \tag{28}$$

where $\mathbf{X}_t := \{S_t^1, S_t^2\}$ is a two-dimensional process, $\mathbf{W}_t := \{W_t^1, W_t^2\}$ is a two-dimensional Brownian motion with instantaneous correlation coefficient between W_t^1 and W_t^2 being ρ, and σ takes values in $\mathbb{R}^2$.

The payoff of the spread option is given by:

$$h(\mathbf{X}_{t_m}) = \max\left(S_{t_m}^1 - S_{t_m}^2 - K, 0\right),$$

where K is the strike of the option. The Bermudan option has eight equally spaced early-exercise opportunities, with the final maturity at $T = 1$ year. The parameters for this case are taken from Set 2 in Table 5.1.1.

Specific to the SGBM case we use the recursive bifurcation scheme for bundling, with 4^3 bundles

	SGBM extended Delta (s.e)	SGBM BR Delta (s.e)	LSMC1 BR Delta (s.e.)	LSMC2 BR Delta (s.e.)
$\frac{\partial V_{t_0}}{\partial S^1_{t_0}}$	0.4020 (0.2e-4)	0.4021 (0.1e-3)	0.4029 (0.011)	0.4570 (0.083)
$\frac{\partial V_{t_0}}{\partial S^2_{t_0}}$	-0.3448 (0.2e-4)	-0.3453 (0.1e-3)	-0.3446 (0.010)	-0.3795 (0.085)

Table 8. t_0 Delta values for Bermudan spread option on two assets. The values in brackets are the standard errors from thirty trials. The parameters for this experiment are taken from Table 5.1.1 Set 2.

	SGBM extended Vega (s.e)	SGBM BR Vega (s.e)	LSMC1 BR Vega (s.e.)	LSMC2 BR Vega (s.e.)
$\frac{\partial V_{t_0}}{\partial \sigma_1}$	20.6082 (0.016)	20.7551 (0.025)	20.4900 (0.124)	20.5136 (0.198)
$\frac{\partial V_{t_0}}{\partial \sigma_2}$	16.8822 (0.013)	17.0611 (0.017)	17.0022 (0.089)	17.1409 (0.155)

Table 9. Vega t_0 values for Bermudan spread option on two assets. The values in brackets are the standard errors from thirty trials. The parameters for this experiment are taken from Table 5.1.1 Set 2.

Case	SGBM extended	SGBM BR	LSMC1 BR	LSMC2 BR
Single Asset (50 monitoring dates)	4.5s	10s	2s	4.2s
Two Asset (8 monitoring dates)	3s	12s	4s	7s

Table 10. The computational time for the different approaches for the single asset and the two asset case. The experiments were run on an Intel Quadcore processor and the code was implemented in Matlab. Note that SGBM extended computes in total 9 million sensitivities for the single asset case and close to 2.9 million sensitivities for the two asset case.

at each time step. As basis functions we use

$$\begin{aligned}
&\phi_k\left(\mathbf{X}_{t_m}(n)\right) = \left(\left(S^1_{t_m}(n) - S^1_{t_m}(n)\right)\right)^{k-1}, k = 1, \dots, 4,\\
&\phi_5\left(\mathbf{X}_{t_m}(n)\right) = \left(\log\left(S^1_{t_m}(n)\right)\right), \phi_6\left(\mathbf{X}_{t_m}(n)\right) = \left(\log\left(S^1_{t_m}(n)\right)\right)^2\\
&\phi_7\left(\mathbf{X}_{t_m}(n)\right) = \left(\log\left(S^2_{t_m}(n)\right)\right), \phi_8\left(\mathbf{X}_{t_m}(n)\right) = \left(\log\left(S^2_{t_m}(n)\right)\right)^2.
\end{aligned}$$

Again the closed-form conditional expectation, as well as the sensitivity of these conditional expectations with respect to the state space and model parameters are known.

Table 8 compares the SGBM, SGBM *BR* and LSMC1 and LSMC2 *BR* Deltas. Note that with the extended SGBM we are able to compute sensitivities along the paths, although we here compare only the t_0 values. The Delta values from the SGBM extended, SGBM *BR* and LSMC1 *BR* are consistent. LSMC2 is slightly off compared to the other three, a behavior consistent with the observations for the single asset case discussed in Section 5.2.

Table 9 reports the corresponding Vega values for the above case.

The results for the SGBM extended, SGBM *BR*, LSMC1 *BR* and LSMC2 *BR* are comparable, although the lowest standard error is for the SGBM extended values. The combined computational time for SGBM where option Deltas and Vegas are computed along the scenarios on all the monitoring dates is less than 3 seconds on an Intel Quadcore processor with 4 GB RAM.

Table 10 summarizes the mean computational time for 30 trials for this experiment.

6. Conclusion

We have presented an approach to compute sensitivities with respect to state space and model parameters along the path for early-exercise options. The approach is applicable to regress-later schemes, in particular, to the considered SGBM. Through basic examples we illustrate the approach, numerically study the convergence of the method and demonstrate the stability of the method. It was found in our numerical experiments that the *BR* LSMC Greeks may be unstable when the initial asset price is close to the early-exercise boundary. The sensitivities along the paths are computed without significant computational and memory overhead. Based on the quality of the Greeks along the paths that are computed at significantly low computational costs, we believe the method is ideal for computing MVA for SIMM based initial margins, where sensitivities along the paths are required to proxy the initial margin along the scenarios.

References

[1] A. Antonov. Algorithmic differentiation for callable exotics. `https://ssrn.com/abstract=2839362`, 2016.

[2] A. Antonov, S. Issakov, and S. Mechkov. Backward induction for future values. *Risk*, page 92, 2015.

[3] E. Beutner, A. Pelsser, and J. Schweizer. Fast convergence of regress-later estimates in least squares Monte Carlo. `https://ssrn.com/abstract=2328709`, 2013.

[4] E. Beutner, A. Pelsser, and J. Schweizer. Theory and validation of replicating portfolios in insurance risk management. `https://ssrn.com/abstract=2557368`, 2016.

[5] M. Broadie and P. Glasserman. A Stochastic Mesh Method for pricing high-dimensional American options. *Journal of Computational Finance*, 7:35–72, 2004.

[6] L. Capriotti, Y. Jiang, and A. Macrina. AAD and least-square Monte Carlo: Fast Bermudan-style options and XVA Greeks. *Algorithmic Finance*, 6(1-2):35–49, 2017.

[7] F. Cong and C. W. Oosterlee. Pricing bermudan options under merton jump-diffusion asset dynamics. *International Journal of Computer Mathematics*, 92(12):2406–2432, 2015.

[8] C. S. De Graaf, Q. Feng, D. Kandhai, and C. W. Oosterlee. Efficient computation of exposure profiles for counterparty credit risk. *International Journal of Theoretical and Applied Finance*, 17(04):1450024, 2014.

[9] Q. Feng, S. Jain, P. Karlsson, D. Kandhai, and C. W. Oosterlee. Efficient computation of exposure profiles on real-world and risk-neutral scenarios for bermudan swaptions. *Journal of Computational Finance*, 20(1):139–172, 2016.

[10] Q. Feng and C. Oosterlee. Calculation of exposure profiles and sensitivities of options under the Heston and the Heston Hull-White models. In *Recent Progress and Modern Challenges in Applied Mathematics, Modeling and Computational Science*, pages 265–301. Springer, 2017.

[11] E. Fournié, J.-M. Lasry, J. Lebuchoux, P.-L. Lions, and N. Touzi. Applications of Malliavin calculus to Monte Carlo methods in finance. *Finance and Stochastics*, 3(4):391–412, 1999.

[12] M. Giles and P. Glasserman. Smoking adjoints: Fast Monte Carlo Greeks. *Risk*, 19(1):88–92, 2006.

[13] P. Glasserman. *Monte Carlo methods in financial engineering*, volume 53. Springer, 2004.

[14] P. Glasserman and B. Yu. Simulation for American options: Regression now or regression later? In *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pages 213–226. Springer, 2004.

[15] E. Gobet, A. Kohatsu-Higa, et al. Computation of Greeks for barrier and lookback options using Malliavin calculus. *Electronic communications in probability*, 8:51–62, 2003.

[16] S. Jain, P. Karlsson, and D. Kandhai. Kva, mind your p's and q's! `https://ssrn.com/abstract=2557368`, 2016.

[17] S. Jain and C. W. Oosterlee. The stochastic grid bundling method: Efficient pricing of bermudan options and their greeks. *Applied Mathematics and Computation*, 269:412–431, 2015.

[18] P. Karlsson, S. Jain, and C. W. Oosterlee. Counterparty credit exposures for interest rate derivatives using the stochastic grid bundling method. *Applied Mathematical Finance*, 23(3):175–196, 2016.

[19] P. Karlsson, S. Jain, and C. W. Oosterlee. Fast and accurate exercise policies for bermudan swaptions in the libor market model. *International Journal of Financial Engineering*, 3(01):1650005, 2016.

[20] A. Kohatsu-Higa and M. Montero. Malliavin calculus in finance. *Handbook of computational and numerical methods in finance*, pages 111–174, 2004.

[21] Á. Leitao and C. W. Oosterlee. GPU acceleration of the stochastic grid bundling method for early-exercise options. *International Journal of Computer Mathematics*, 92(12):2433–2454, 2015.

[22] A. Leitao Rodriguez, C. W. Oosterlee, L. Ortiz-Gracia, and S. Bohte. On the data-driven cos method. https://ssrn.com/abstract=2917536, 2017.

[23] P. Letourneau and L. Stentoft. Improved Greeks for American options using simulation. 2016.

[24] F. A. Longstaff and E. S. Schwartz. Valuing American options by simulation: A simple least-squares approach. *Review of Financial studies*, 14(1):113–147, 2001.

[25] A. Pelsser and J. Schweizer. The difference between LSMC and replicating portfolio in insurance liability modeling. *European Actuarial Journal*, 6(2):441–494, 2016.

[26] M. Ruijter and C. Oosterlee. Two-dimensional Fourier cosine series expansion method for pricing financial options. *SIAM Journal on Scientific Computing*, 34(5):B642–B671, 2012.

[27] Y. Wang and R. Caflisch. Pricing and hedging American-style options: a simple simulation-based approach. *Journal of Computational Finance*, 13(4):95, 2010.

Appendix A: On Regress-later and bundling

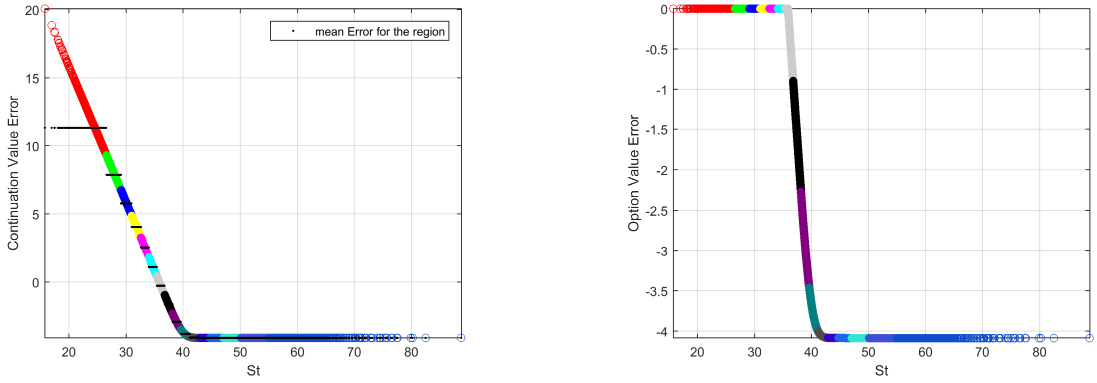
The major challenge for regress-later or portfolio replication schemes arises for derivatives that exhibit strong path dependency, which, as Pelsser and Schweizer (2016)(25) demonstrate, often implies a higher problem dimensionality. Finding a suitable regression basis for pricing such derivatives, especially one for which the conditional expectations are readily available is therefore challenging.

The common challenge for partitioning-based schemes is to determine the optimal way to partition a high-dimensional state space. As usual, a suitable approximation by partition-based schemes is obtained with a large number of partitions, however, these methods tend to be computationally intensive. Computing sensitivities along the paths using basic partitioning schemes is also non-trivial.

In SGBM, as for each partition a functional approximation of the future option value is obtained for which the conditional expectation is known in closed-form, the need for a rigorous approach to optimally partition the state space is not essential. Also, because the conditional distribution is sampled directly, by using the bundling scheme in SGBM, it is not essential to define an excellent regression fit for the future option value. In short, SGBM benefits from best of both, pure partitioning based schemes and regress-later schemes.

To illustrate the bias that affects a pure regress-later scheme when the set of basis functions is not chosen appropriately, we consider the Bermudan option on a single asset from Section 5.2. We consider the case where the only basis function is a constant, i.e. $\Phi_1(\mathbf{X}_{t_m}) = 1$. We focus on computing the continuation value at t_{M-1}. The paths are divided into partitions and we look at the residual error of the continuation and option values both globally, i.e. combined error of all the paths, and locally, i.e. the aggregated error for each of the partition. Note that at this time step the continuation value is known in closed-form, which is equal to the BS price for a European option at t_{M-1} which expires in $\Delta t = t_M - t_{M-1}$. The closed-form option values are computed as the maximum of the continuation and intrinsic value at this time step.

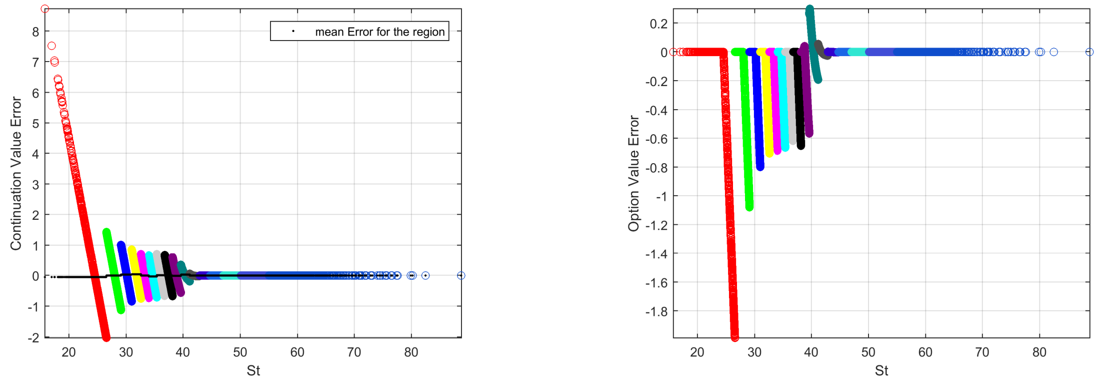
In Figure A1 the error in the continuation and option values for the simulated states at t_{M-1} for the basic regress-later scheme are displayed. Different colors correspond to different partitions, where for our analysis the errors are locally aggregated. Regarding the continuation value, although the combined residual error for all regions is close to zero, the local errors can be significant. With the option value the maximum of the conditional expectation and the intrinsic value, non-zero local errors add to the bias, which is visible in the error plot for the option values.



(a) (b)

Figure A1. The error in (a) continuation values and (b) option value at t_{M-1} for a basic regress-later scheme, the different colors indicate different regions for evaluation of local residual errors.

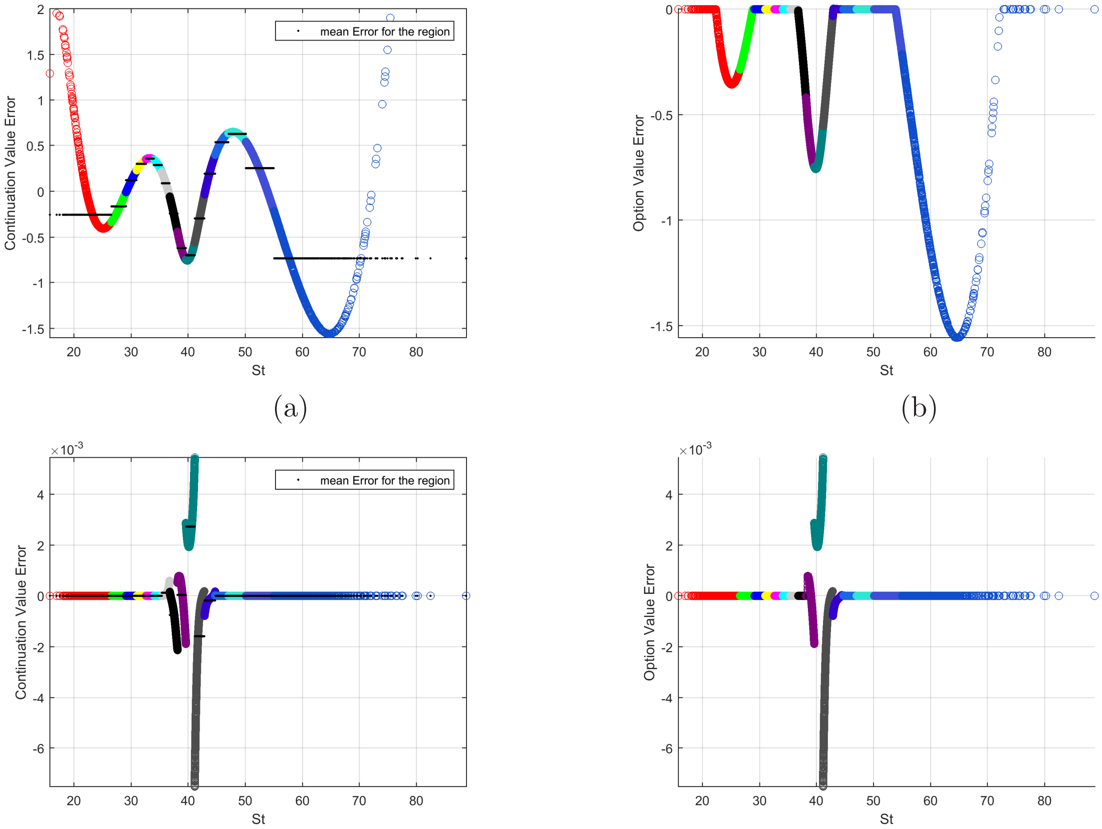
We perform the same exercise for SGBM, where the constant is again the only basis vector for regression. The local residual errors for the different partitions are displayed in Figure A2. With bundling the mean of both the local and global residual errors are close to zero. For path-dependent options it is important that the local residual error of the conditional expectation to minimize the impact of wrong early-exercise decisions.



(a) (b)

Figure A2. The error in (a) continuation values and (b) option value at t_{M-1} for SGBM. The different colors indicate different regions for evaluation of local residual errors.

The squared residual error from regression in regress-later schemes gets closer to zero with a richer set of basis functions. As the squared residual error gets smaller (approaches zero) the error in the continuation value approaches zero, both globally and in the local regions, for a basic regress-later scheme. This can be seen in Figure A3 where we use the basis functions as defined in Equation (26) with $K = 6$. However, finding basis functions such that the squared residual is close to zero, with the additional constraint of closed-form conditional expectations of these functions, is not always straightforward. Even for a case where a good set of basis functions is used, bundling can additionally reduce the errors. This can be seen in Figure A3 (c) and (d), where for the same set of basis functions, 2^4 bundles are used within SGBM.



(a)

(b)

(c)

(d)

Figure A3. The error in (a) continuation values and (b) option value at t_{M-1} for a pure regress-later scheme when as basis power upto 5 of log of asset price is used. The correspnding errors for SGBM with 16 bundles are in (c) and (d) respectively. Different colors indicate different regions for evaluation of local residual errors.